

SPESISIFIKASI SISTEM *AUTOMATIC*
VEHICLE CLASSIFICATION MENGGUNAKAN
B-METHOD



Studi Mandiri Software Engineering

Bayu Tenoyo

1006836604

Fakultas Ilmu Komputer

Universitas Indonesia

Depok 2011

Abstrak

Automatic Vehicle Classification merupakan sebuah sistem terpadu yang terdiri dari perangkat keras dan perangkat lunak, untuk digunakan mengklasifikasikan secara otomatis kendaraan yang melewati sebuah gerbang tol sesuai dengan golongannya. Golongan yang ada telah ditentukan berdasarkan hubungan antara variabel tertentu, misalkan dalam kasus ini melihat jumlah gandar, jumlah roda, apakah bis atau bukan.

Laporan studi mandiri ini akan menjelaskan spesifikasi yang diperlukan untuk mengembangkan sistem *Automatic Vehicle Classification* dengan menggunakan B Method, sehingga memungkinkan membuktikan kebenaran spesifikasi tersebut dengan logika matematika, dalam hal ini menggunakan sebuah alat bantu: Atelier B.

1. Pendahuluan

Menghasilkan spesifikasi sistem tertanam (*embedded system*) yang berkualitas dalam arti tidak ambigu dan lengkap merupakan hal yang susah. Kesalahan spesifikasi sering terdeteksi saat dilakukan integrasi terakhir dan sistem testing. Padahal melakukan perbaikan kesalahan pada tahapan sistem testing memerlukan biaya yang 10 sampai 1000 kali dibandingkan melakukan perbaikan tersebut pada tahapan spesifikasi [7].

Sistem *Automatic Vehicle Classification* merupakan sistem tertanam yang digunakan untuk menggolongkan kendaraan yang masuk dalam sebuah gardu tol sesuai dengan relasi atau aturan antar variabel yang sudah ditentukan. Contoh variabel ini antara lain state dari sensor-sensor yang ada dalam sistem.

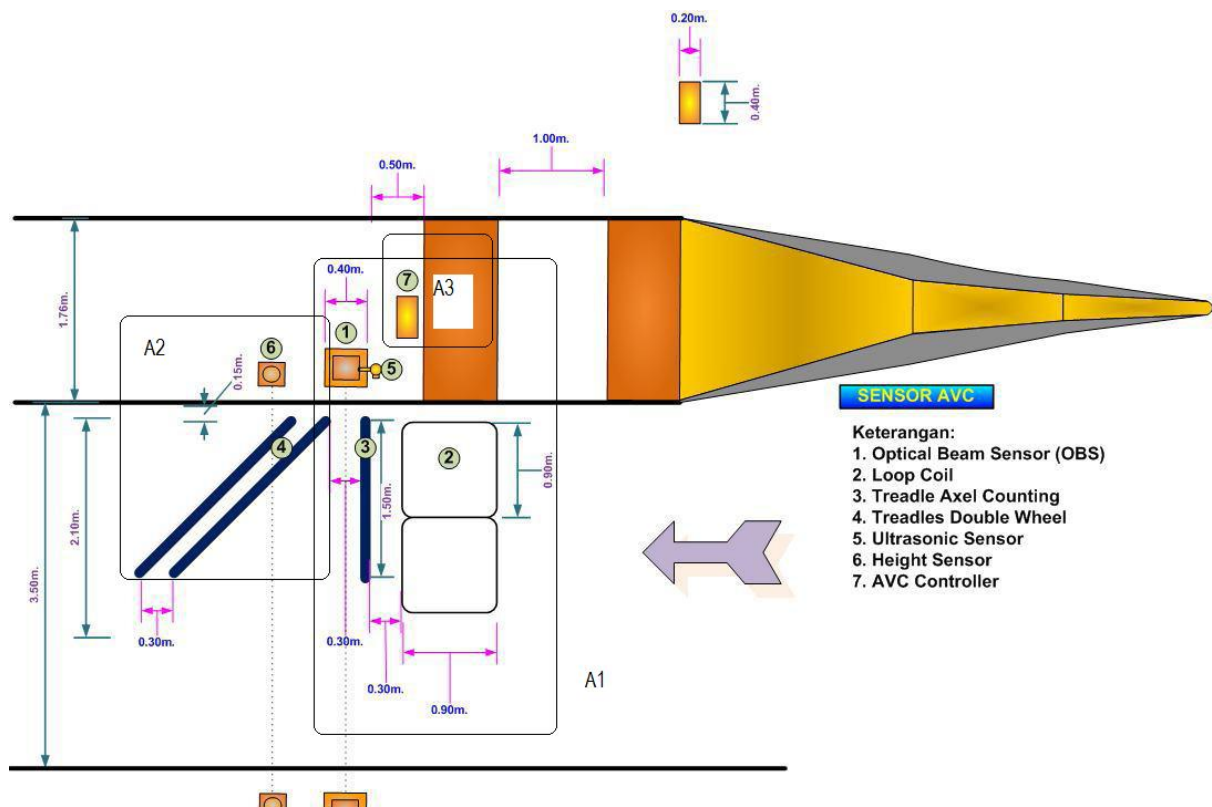
B merupakan metoda untuk menspesifikasikan, mendisain, dan mengembangkan kode program dari sebuah sistem. Landasan utamanya adalah teori himpunan dari Zermelo-Fraenkel dengan *axiom of choice*, penggunaan konsep substitusi umum pada mekanisme terstruktur dalam hal ini *machine*, *refinement*, dan *implementation* [2]. *Machine* merupakan mekanisme terstruktur digunakan untuk menyatakan spesifikasi, sedangkan *refinement* dan *implementation* merupakan kelanjutan dari spesifikasi yang sudah dibuat dengan hasil akhir kode program. Dalam laporan tugas mandiri ini, kita hanya akan menggunakan B untuk menyatakan spesifikasi dan membuktikan spesifikasi tersebut benar dengan alat bantu Atelier B.

B-Method sendiri sudah digunakan untuk berbagai macam solusi, diantaranya adalah: *flight control system* [7], sebagai *embeded verifier* untuk *java card byte code* [6], *automatic train protection* [8], *automatic train control* – tanpa pengemudi [8], fungsional dari subsistem di mobil seperti: lampu, air bag, mesin, dll [8], pemodelan pintu otomatis untuk jalur kereta api [8].

Atelier B, perangkat lunak yang dikembangkan oleh ClearSy System Engineering untuk mendukung B-Method. Alat ini memungkinkan kita untuk membuktikan kebenaran sebuah mekanisme terstruktur apakah itu *machine*, *refinement*, dan *implementation*, terhadap sifat-sifat sistem, maupun keterhubungan antara ketiganya.

Laporan akhir ini akan diorganisasikan sebagai berikut, gambaran singkat tentang Sistem *Automatic Vehicle Classification* pada Bab 2, teori singkat mengenai B-Method pada Bab 3, spesifikasi *Automatic Vehicle Classification* pada Bab 4, dan pemakaian Atelier-B pada Bab 5.

2. Sistem *Automatic Vehicle Classification*



Gambar 1. *Automatic Vehicle Classification* Sensors

Sistem *automatic vehicle classification* ini diinstal pada gerbang tol otomatis, dalam artian bahwa pengemudi mendapatkan kartu masuk jalan tol bukan dari petugas melainkan dari sebuah dispenser. Dimana kartu yang diterima oleh pengemudi sesuai dengan golongan kendaraannya, penggolongan tersebut disimpulkan dari hasil yang dikirim sensor-sensor yang ada pada sistem.

Dari Gambar 1 (mengacu pada referensi [9]) bisa dilihat bahwa *automatic vehicle classification* terdiri dari enam buah sensor dan sebuah controller. Sensor tersebut dan kegunaannya sebagai berikut:

- *Optical beam sensor*, digunakan untuk mengetahui kendaraan ada atau tidak ada, selain itu sebagai tanda pemisah antara kendaraan yang di depan dengan yang di belakangnya.
- *Loop coil*, mendeteksi apakah ada kendaraan di atasnya.
- *Treadle axel counting*, menghitung banyaknya gandar yang dimiliki oleh sebuah kendaraan.
- *Ultrasonic sensor*, mengetahui apakah kendaraan bis ataukah bukan.

- *Height sensor*, mendeteksi tinggi kendaraan (dalam spesifikasi kali ini hasil dari sensor ini belum digunakan).
- *AVC controller*, untuk menyimpulkan golongan apakah kendaraan yang melalui gerbang tol ini.

Untuk saat ini dikenal 5 golongan kendaraan dimana setiap golongan mempunyai tarif yang berbeda-beda. Aturan penggolongan kendaraannya adalah sebagai berikut:

- Golongan 1 – aturan 1 : banyaknya gandar 2, dan tidak *double wheel* (mobil) .
- Golongan 1 – aturan 2 : banyaknya gandar 2, *double wheel*, dan kendaraan adalah bis.
- Golongan 2: banyaknya gandar 2, *double wheel*, bukan bis.
- Golongan 3: banyaknya gandar 3.
- Golongan 4: banyaknya gandar 4.
- Golongan 5: banyaknya gandar 5.

3. B Method – *Abstract Machine*

Untuk menyatakan spesifikasi suatu sistem, dalam B kita bisa menyatakannya dalam sebuah *Abstract Machine* [1]. Dalam sebuah *abstract machine* kita tidak menggunakan *sequencing* dan *looping* didalamnya, karena kita membuat abstract machine bukan untuk mengembangkan kode program tetapi lebih membuat spesifikasi dari sistem.

Notasi yang digunakan sering disebut *Abstract Machine Notation* (AMN). Fitur-fitur yang ada dalam notasi AMN antara lain: *pre-condition*, *multiple assignment*, *bounded choice*, *guard*, dan *unbounded choice*.

Struktur dari sebuah *abstract machine* yang akan digunakan dalam laporan studi mandiri ini dapat dilihat sebagai berikut (bentuk lebih lengkapnya dapat dilihat pada referensi-referensi yang ada [1][5]):

```

MACHINE
  nama_mesin

CONSTANTS
  constant1, constant2, ..., constantm

PROPERTIES
  ppredicate1 ∧ ppredicate2 ∧ ... ∧ ppredicaten

DEFINITIONS
  definition1; definition2; ...; definitiono

SETS
  SET1; SET2; ...; SETp

VARIABLES

```

```

variable1, variable2, ..., variableq

INVARIANT
  ipredicate1 ∧ ipredicate2 ∧ ... ∧ ipredicater

INITIALISATION
  init1 || init2 || ... || inits

OPERATIONS
  operation1 ≙
  ....
  END;
  .....

  outvalue ← operationt(param1, param2) ≙
  ....
  END

END

```

Penjelasannya sebagai berikut:

- nama_mesin: merupakan nama *abstract machine* yang dispesifikasikan.
- constant₁, constant₂, ..., constant_m: merupakan daftar konstanta yang berlaku dalam *abstract machine* nama_mesin.
- ppredicate₁ ∧ ppredicate₂ ∧ ... ∧ ppredicate_n: adalah daftar properti yang berlaku dalam nama_mesin, dinyatakan dalam predikat. Predikat-predikat yang ada menjelaskan batasan yang berlaku untuk konstanta-konstanta yang sudah didefinisikan, dalam nama_mesin.
- definition₁; definition₂; ...; definition_o: adalah daftar definisi yang berlaku pada nama_mesin, mirip peran makro pada bahasa C. Sebagai contoh: INDEX_MBL == 1..1000, dengan adanya definisi tersebut, setiap kemunculan INDEX_MBL pada klausa-klausa yang lain (misalkan pada klausa *OPERATIONS*) sebenarnya menggantikan 1..1000.
- SET₁; SET₂; ...; SET_p: adalah definisi kumpulan himpunan yang digunakan dalam nama_mesin.
- variable₁, variable₂, ..., variable_q: adalah daftar variabel yang digunakan dalam nama_mesin.
- ipredicate₁ ∧ ipredicate₂ ∧ ... ∧ ipredicate_r: adalah kumpulan predikat yang menjadi batasan variabel-variabel, yang berlaku dalam nama_mesin.
- init₁ || init₂ || ... || init_s: merupakan pemberian nilai awal dari nilai variabel-variabel yang ada, nilai tersebut berlaku sampai variabel bersangkutan mendapatkan assignment dalam klausa *OPERATIONS*.
- operation₁, ..., operation_t: merupakan spesifikasi operasi-operasi yang berlaku dalam nama_mesin. Sebuah operasi bisa dibayangkan seperti *method* pada paradigma *Object Oriented Programming*, dan sebuah variabel seperti *attribute*. Variabel-variabel yang

berlaku pada *abstract machine* hanya bisa dirubah dan diakses nilainya melalui sebuah operasi.

`outvalue` merupakan output parameter dari `operationt`, seperti *return value* dari sebuah *method*.

`param1`, `param2` merupakan input parameter dari , seperti input parameter dari sebuah *method*.

4. Spesifikasi *Automatic Vehicle Classification*

Spesifikasi utama dalam AVC adalah:

- Kendaraan yang masuk harus masuk dalam golongan yang sudah ditentukan.
- Kendaraan yang masuk memperoleh tiket/kartu yang didalamnya terdapat informasi golongannya.

Dalam B, spesifikasi tersebut dapat dijabarkan sebagai berikut:

- Nama mesin yang digunakan dalam spesifikasi ini adalah `avc`.

```
MACHINE
  avc
  ....
END
```

- Daftar konstanta yang ada dalam sistem adalah:
 - `maxgandar`, merepresentasikan jumlah gandar maksimum yang bisa diterima oleh sistem. Dalam *abstract machine* nilai konkrit dari konstanta tidak harus dinyatakan.
 - `maxmobil`, merepresentasikan jumlah maximum mobil yang dapat disimpan dalam sistem.

```
MACHINE
  avc

  CONSTANTS
    maxgandar, maxmobil
    ....

END
```

- Predikat yang harus berlaku atas konstanta-konstanta di atas adalah:
 - `maxgandar` adalah anggota bilangan natural yang dimulai dari 1, **dan**
 - `maxmobil` adalah anggota bilangan `1..MAXINT`.

```
MACHINE
  avc

  CONSTANTS
    maxgandar, maxmobil
```

```

PROPERTIES
  maxgandar ∈ NATURAL1 ∧
  maxmobil ∈ 1..MAXINT-1

  ....

END

```

- Definisi yang ada adalah:
 - INDEXMBL == 1..maxmobil.

```

MACHINE
  avc

CONSTANTS
  maxgandar, maxmobil
  ....

DEFINITIONS
  INDEXMBL == 1..maxmobil
  ....

END

```

- Himpunan yang ada dalam sistem:
 - GOLONGAN, himpunan yang merepresentasikan golongan-golongan yang ada dalam sistem.
 - STATUS, himpunan yang merepresentasikan lokasi sebuah mobil terhadap daerah A1 (lihat Gambar 1).
 - KARTUSTS, himpunan yang merepresentasikan status kartu untuk sebuah mobil yang sudah memasuki GTO dan sudah.

```

MACHINE
  avc

CONSTANTS
  maxgandar, maxmobil
  ....

SETS
  GOLONGAN = {Golongan1, Golongan2,
  Golongan3, Golongan4, Golongan5};
  STATUS = {belumlewat, didalam, keluar};
  KARTUSTS={notavailable, available}
  ....

END

```

- Variabel-variabel yang ada dalam sistem:
 - mobilDiA1, menyimpan status lokasi dari kendaraan dilihat dari area A1, apakah belum lewat, ada di dalam, ataukah di luar dari A1.

- benarbis, jika nilainya TRUE kendaraan adalah sebuah bis, FALSE jika bukan.
 - axlenb, menyimpan nilai banyaknya gandar yang terdeteksi dari kendaraan.
 - benardouble, jika nilainya TRUE artinya ada roda ganda untuk satu sisi dari sebuah gandar (empat roda dalam satu gandar), FALSE jika tidak benar.
 - treadledoublewpress, menyimpan jumlah gandar yang mempunyai roda ganda.
 - mblgl, menyimpan golongan dari sebuah kendaraan.
 - imblgl, indeks dari sebuah kendaraan, menunjukkan urutan masuknya kendaraan ke dalam gerbang tol bersangkutan.
 - golongansudahdiset, untuk mengetahui apakah golongan untuk sebuah kendaraan sudah ditentukan atau belum.
- Invarian, atau kondisi yang harus selalu terpenuhi untuk setiap variabel dalam `avc` adalah:
 - `mobildiA1` anggota himpunan STATUS, nilainya adalah anggota dari {belumlewat, didalam, keluar}, **dan**
 - `benarbis` anggota himpunan BOOL, nilainya TRUE atau FALSE, **dan**
 - `axlenb` anggota himpunan {0,1,..., maxgandar}, **dan**
 - `benardouble` anggota himpunan BOOL, **dan**
 - `treadledoublewpress` anggota himpunan {0,1,...,maxgandar}, **dan**
 - `mblgl` anggota himpunan dari total fungsi dengan domain {1,2,...,maxmobil} dan range {Golongan1,Golongan2,Golongan3,Golongan4,Golongan5}, artinya semua kendaraan yang tercatat memasuki gerbang tol akan dipetakan ke dalam golongannya, **dan**
 - `imblgl` anggota himpunan {1,2,...,maxmobil+1}, **dan**
 - `golongansudahdiset` anggota himpunan BOOL.

MACHINE

`avc`

CONSTANTS

`maxgandar, maxmobil`

.....

VARIABLES

`mobildiA1, benarbis, axlenb, benardouble, treadledoublewpress, mblgl,mblkrt, imblgl, imblkrt, golongansudahdiset`

INVARIANT

`mobildiA1 ∈ STATUS ∧
 benarbis ∈ BOOL ∧
 axlenb ∈ 0..maxgandar ∧
 benardouble ∈ BOOL ∧
 treadledoublewpress ∈ 0..maxgandar ∧
 mblgl ∈ INDEXMBL → GOLONGAN ∧
 mblkrt ∈ INDEXMBL → KARTUSTS ∧
 imblgl ∈ 1..maxmobil+1 ∧
 imblkrt ∈ 0..maxmobil ∧
 (imblkrt>0⇒mblkrt(imblkrt)=available) ∧`


```

golongsudahdiset ∈ BOOL
....
END

```

- Inialisasi dari variabel-variabel dinyatakan sebagai berikut:

```

MACHINE
  avc

CONSTANTS
  maxgandar, maxmobil
  ....

INITIALISATION
  mobildiA1:=belumlewat ||
  benarbis:=FALSE ||
  axlenb:=0 ||
  treadledoublewpress:=0 ||
  mblgl:: INDEXMBL → GOLONGAN ||
  mblkrt:: INDEXMBL → KARTUSTS ||
  imblgl:=1 ||
  imblkrt:=0 ||
  golongsudahdiset:=FALSE
  ....

END

```

- Berikut ini adalah operasi-operasi yang harus ada dalam sistem:
 - Menandakan sebuah kendaraan itu bis atau bukan, prekondisi yang harus terpenuhi adalah kendaraan ada di area A1 atau sudah keluar dari area A1, dan sensor ultrasonic memberikan tanda bahwa kendaraan yang terdeteksi adalah bis.

```

MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic)  $\hat{=}$ 
  PRE
    (mobildiA1=didalam or mobildiA1=keluar)  $\wedge$ 
    benarbis=FALSE  $\wedge$ 
    fromUltrasonic=TRUE
  THEN
    benarbis:=TRUE
  END;
  ....

END

```

- Menghitung gandar dari sebuah kendaraan, dengan prekondisi yang harus dipenuhi: kendaraan masih dalam area A1, dan sensor treadle axle counting memberikan

sinyal bahwa gandar baru terdeteksi, dan jumlah gandar yang terdeteksi tidak melebihi jumlah maksimum gandar yang ditetapkan.

```
MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic) ≐
  ....
  END;

  inctraxlecounting(treadleaxle) ≐
  PRE
    mobildiA1=didalam ∧
    treadleaxle=TRUE ∧
    axlenb<maxgandar
  THEN
    axlenb:=axlenb+1
  END;

  ....

END
```

- o Menentukan apakah sebuah kendaraan sudah memasuki area A1, ataukah sudah keluar. Informasi lokasi kendaran ini digunakan pada operasi-operasi yang lain: sekiranya kendaraan tersebut sudah keluar untuk menandakan bahwa penghitungan gandar terhadap kendaraan tersebut sudah selesai, sekiranya kendaraan tersebut di dalam A1 berarti penghitungan gandar dimulai atau bisa dilanjutkan. Kendaraan di dalam A1 sekiranya salah satu sensor berikut bernilai TRUE : *loop coil*, *ultrasonic*, atau *optical beam*. Sedangkan kendaraan dianggap keluar dari A1 jika ketiga sensor tersebut menghasilkan nilai FALSE.

```
MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic) ≐
  ....
  END;

  ....
  posisidiA1(loopcoil,ultrasonic,opticalbeam)
  PRE
    loopcoil ∈ BOOL ∧
    ultrasonic ∈ BOOL ∧
    opticalbeam ∈ BOOL
```

```

THEN
  select loopcoil=TRUE ∨ ultrasonic=TRUE ∨
         opticalbeam=TRUE THEN
    mobildiA1:=didalam
  when loopcoil=FALSE ∧ ultrasonic=FALSE ∧
         opticalbeam=FALSE ∧ mobildiA1=didalam
  THEN
    mobildiA1:=keluar
  END
END;

....

END

```

- Mendeteksi apakah ada gandar dengan roda ganda. Prekondisi yang harus dipenuhi adalah: kendaraan ada di dalam A1 atau sudah keluar dari A1, treadle double wheel sensor memberitahu adanya gandar dengan roda ganda, selain itu pengecekan dilakukan jika *treadle double wheel* sudah tertekan lebih besar atau sama dengan 2.

```

MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic) ≐
  ....
  END;
  ....
  posisidiA2(tdwpress, isdouble) ≐
  PRE
    (mobildiA1=didalam ∨ mobildiA1=keluar) ∧
    tdwpress=TRUE ∧
    isdouble ∈ BOOL ∧
    treadledoublewpress ≥ 2
  THEN
    if isdouble=TRUE THEN
      benardouble:=TRUE
    END;
  ....

END

```

- Menghitung jumlah gandar yang terdeteksi oleh *treadle double wheel* sensor. Terlepas apakah itu roda ganda atau tidak. Prekondisi yang terpenuhi adalah kendaraan di dalam A1 atau sudah keluar di A1 dan sensor *treadle double wheel* tertekan.

```

MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic)  $\hat{=}$ 
  ....
  END;
  ....
  inctreadledoublew(tdwpress)  $\hat{=}$ 
  PRE
    (mobildiA1=didalam  $\vee$  mobildiA1=keluar)  $\wedge$ 
    tdwpress=TRUE  $\wedge$ 
    treadledoublewpress<maxgandar
  THEN
    treadledoublewpress:=treadledoublewpress+1
  END;
  ....

END

```

- Variabel-variabel yang digunakan untuk mengolah sebuah kendaraan, harus segera direset, setelah kendaraan tersebut sudah ditentukan golongannya. Sehingga variabel-variabel tersebut bisa digunakan untuk merepresentasikan state dari kendaraan selanjutnya (yang ada dibelakangnya). Prekondisi yang harus terpenuhi adalah kendaraan sudah keluar dari area A1 dan golongan kendaraan sudah ditentukan.

```

MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic)  $\hat{=}$ 
  ....
  END;
  ....
  resetvariables  $\hat{=}$ 
  PRE
    mobildiA1=keluar  $\wedge$ 
    golongan sudahdiset=TRUE
  THEN
    mobildiA1:=belumlewat || benarbis:=FALSE ||
    axlenb:=0 || treadledoublewpress:=0 ||
    golongan sudahdiset:=FALSE
  END;
  ....

END

```

- o Menentukan golongan dari kendaraan menggunakan aturan yang ada. Prekondisi yang harus dipenuhi adalah: kendaraan sudah keluar dari A1 dan indeks kendaraan tidak melebihi dari maksimum kendaraan yang dapat dicatat dalam sistem.

```

MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic) ≐
  ....
  END;
  ....
  menentukanmblgl ≐
  PRE
    mobildiA1=keluar ∧
    imblgl < maxmobil
  THEN
    select axlenb=2 ∧ benardouble=FALSE THEN
      mblgl <+ {imblgl ↦ Golongan1}
    when axlenb=2 ∧ benardouble=TRUE ∧
      benarbis=TRUE THEN
      mblgl <+ {imblgl ↦ Golongan1}
    when axlenb=2 ∧ benardouble=TRUE ∧
      benarbis=FALSE THEN
      mblgl <+ {imblgl ↦ Golongan2}
    when axlenb=3 THEN
      mblgl <+ {imblgl ↦ Golongan3}
    when axlenb=4 THEN
      mblgl <+ {imblgl ↦ Golongan4}
    when axlenb>=5 THEN
      mblgl <+ {imblgl ↦ Golongan5}
    END ||
    imblgl:=imblgl+1 ||
    golongan sudahdiset:=TRUE
  END;
  ....

END

```

- o Mengembalikan nilai ke nilai inisialisasi dari variabel yang merepresentasikan jumlah kendaraan yang dicatat dalam sistem, sekiranya nilai variabel tersebut sudah lebih batas maksimum.

```

MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic) ≐
  ....
  END;
  ....
  resetindexmobilgol ≐
  PRE
    imblgl = maxmobil + 1
  THEN
    imblgl:= 1
  END
  END;
  ....

END

```

- Mengembalikan nilai ke nilai inisialisasi dari variabel yang merepresentasikan jumlah kendaraan yang sudah menerima kartu yang dicatat dalam sistem, sekiranya nilai variabel tersebut sudah melebihi batas maksimum.

```

MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic) ≐
  ....
  END;
  ....
  resetindexmobilkartu ≐
  PRE
    imblkrt = maxmobil
  THEN
    imblkrt:= 0
  END
  END;
  ....

END

```

- Menyediakan golongan untuk kendaraan yang sudah digolongkan dan memberi tanda bahwa kartu untuk kendaraan tersebut sudah bisa diambil. Prekondisi yang harus terpenuhi adalah: kendaraan yang akan mengambil kartu golongannya sudah ditentukan, dan kendaraan sebelumnya sudah mengambil kartu, dan indeks yang

merepresentasikan kendaraan yang sudah mengambil kartu tidak melewati batas maksimum yang sudah ditentukan.

```

MACHINE
  avc
  ....

OPERATIONS
  setbenarbis(fromUltrasonic) ≐
  ....
END;
  ....
  valgolongan<--ambilkartumobil =
  PRE
    golongan sudahdiset=TRUE &
    (imblkrt=0 or (mblkrt(imblkrt)=available & imblkrt>0)) &
    imblkrt+1 <= maxmobil
  THEN
    valgolongan:=mblgl(imblkrt+1)||
    imblkrt:=imblkrt+1||
    mblkrt(imblkrt+1):=available
  END;
  ....
END

```

5. Atelier B

Bantuan yang diberikan oleh Atelier B dalam pembuktian spesifikasi sebuah abstract machine ada dua macam, yaitu: *type check* dan *proof obligation*.

Type check dilakukan di awal sebelum kita membuktikan *proof obligation* (PO). Dalam tahapan ini Atelier B akan mencari kesalahan sintaks yang mungkin terjadi dalam mekanisme terstruktur kita, apakah *abstract machine*, ataukah *refinement*, atau *implementation*, dalam hal ini spesifikasi dalam *abstract machine avc*.

Setelah spesifikasi terbebas dari kesalahan sintaks, kita bisa melangkah menghasilkan PO, untuk kemudian dibuktikan kebenarannya. Berikut ini adalah sebuah PO yang diambil dari operasi *inctraxlecounting*:

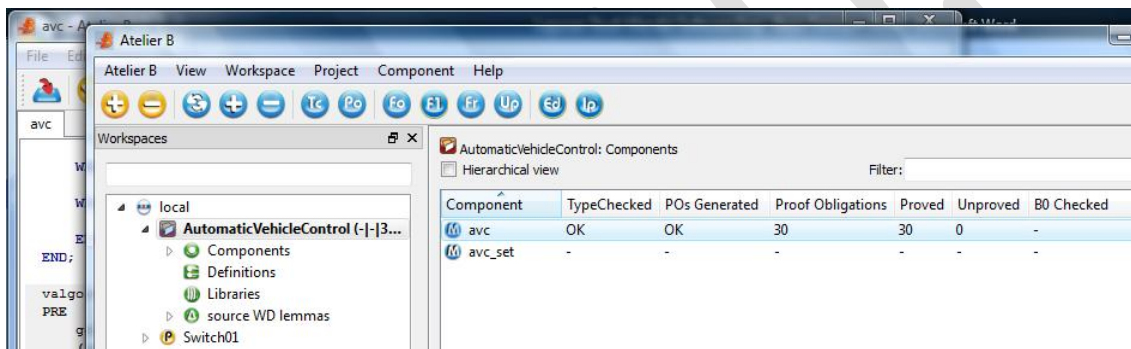
$$axlenb \in 0..maxgandar \Rightarrow axlenb+1 \in 0..maxgandar$$

Banyaknya PO yang ditemukan oleh Atelier B dalam spesifikasi ini adalah sebanyak 30.

No	Klausa/Operasi	Banyaknya PO
1	Initialisation	3
2	setbenarbis	0
3	inctraxlecounting	1

No	Klausa/Operasi	Banyaknya PO
4	posisiA1	0
5	posisiA2	0
6	inctreadledoublew	1
7	resetvariables	1
8	menentukanmblgl	18
9	getgolmobilke	0
10	resetindexmobilgol	1
11	resetindexmobilkrt	1
12	ambilkartumobil	4

Untuk membuktikan semua PO terkadang kita tidak bisa mengandalkan bantuan Atelier B, untuk itu kita bisa memberikan bantuan atau melakukan pembuktian secara interaktif. Pada kasus ini, spesifikasi yang dibuat berhasil dibuktikan secara otomatis. Berikut *screen capture* dari hasil pemeriksaan sintaks dan pembuktian PO secara otomatis.



Gambar 2. Screen Capture Pembuktian avc

Daftar Pustaka

- [1] J.R. Abrial. *The B Book - Assigning Programs to Meanings*. Cambridge University Press, 1996. ISBN 0-521-49619-5.
- [2] Dominique Cansell, Dominique Méry. Foundations of B Method. *Computing and Informatics*, Vol. 22, 2003, 1–31.
- [3] J.R. Abrial. *Formal Method: Theory Becoming Practice*. *Journal of Universal Computer Science*, vol. 13, no. 5 (2007), 619-628.
- [4] *Atelier B*. User Manual Version 4. ClearSy System Engineering.
- [5] *B Language Reference Manual Version 1.8.5*. ClearSy System Engineering.
- [6] Ludovic Casset. *Development of an Embedded Verifier for Java Card Byte Code using Formal Methods*. Gemplus Research Laboratory.
- [7] Liu Jiufu, Yang Zhong. *UML and B Method Based Analysis and Refinement for Flight Control Software of Unmanned Aerial Vehicle*. 2008 International Conference on Computer Science and Software Engineering.
- [8] *Industrial Use of B Method*. ClearSy System Engineering.
- [9] ...Laporan studi mandiri lis...