BAB III

BAHASA MATEMATIS DALAM ILMU KOMPUTER

3.1. BAHASA UNTUK PEMROGRAMAN

3.1.1. Tipe Data Matematis

Dalam bahasa pemrograman baik mulai dari tingkat sederhana sampai tingkat tinggi akan mempunyai kesamaan, yaitu untuk memanipulasi data. Pada bahasa pemrograman yang akan dibahas, hanya mengolah satu tipe data, yaitu berupa himpunan. Tipe data tersebut dapat dikerjakan suatu operasi matematika didalamnya dan merupakan suatu himpunan yang dilengkapi dengan struktur tambahan seperti berikut :

Pandang himpunan A dan

i. fungsi-fungsi $f : A^n \longrightarrow A$, untuk n = 1,2,3,...

ii. predikat-predikat $p : A^n \longrightarrow \{T,F\}$, untuk n = 1,2,3,...

iii. konstan-konstan $c \in A$.

Tipe data didefinisikan sebagai berikut :

$$R = \langle A,\ f_1,\ldots,f_n,\ p_1,\ldots,p_m,\ c_1,\ldots,c_r \rangle$$

dengan ketentuan

A : himpunan yang tidak kosong sebagai *semesta pembicaraan*

f : fungsi

p : predikat

c : konstan

contoh 3.1.1.1

Diberikan tipe data $\mathcal{Z} = \langle Z, +, -, *, <, =, 0, 1 \rangle$, dengan ketentuan

Z : himpunan bilangan bulat

+,-,* : fungsi-fungsi yang didefinisikan berikut :

$+ \longrightarrow f(x,y) = x+y$

$- \longrightarrow f(x,y) = x-y$

$* \longrightarrow f(x,y) = x*y$

<,= : predikat-predikat yang didefinisikan berikut :

$< \longrightarrow p(x,y) =$ "x<y"

predikat ini untuk menguji apakah x<y, bila x<y, maka $p(x,y)$=T dan bila x≥y, maka $p(x,y)$=F

$= \longrightarrow p(x,y) =$ "x=y"

predikat ini untuk menguji apakah x=y, bila x=y, maka $p(x,y)$=T dan bila x≠y, maka $p(x,y)$=F

0,1 : konstan.

3.1.2 Bahasa Pada Suatu Tipe Data

Dalam bahasan ini diawali dengan suatu bahasa yang mempunyai tipe data R sebagai berikut :

Diberikan tipe data

$$R = \langle A, f_1^R, \ldots, f_n^R, p_1^R, \ldots, p_m^R, c_1^R, \ldots, c_r^R \rangle.$$

dan disusun himpunan V yang beranggotakan :

- simbol-simbol variabel individu : $x, y, z, \ldots, x_1, x_2, x_3, \ldots$
- simbol-simbol fungsi : untuk masing-masing fungsi f_i^R adalah fungsi f_i
- simbol-simbol predikat : untuk masing-masing predikat p_i^R adalah predikat p_i

- simbol-simbol konstan : untuk masing-masing konstan c_i^R adalah konstan $\underline{c}_i$
- simbol-simbol punctuasi : $\underline{(}$, $\underline{)}$ dan $\underline{,}$
- simbol-simbol khusus : tergantung bahasa yang akan didefinisikan misal <u>if</u>, <u>then</u>, <u>else</u>, <u>begin</u> dan <u>end</u>.

Simbol-simbol variabel individu digunakan untuk menunjukkan elemen-elemen individu dari himpunan A. Himpunan yang beranggotakan simbol- simbol variabel individu disebut IVS. Masing-masing $\underline{f}_i$ adalah suatu simbol formal yang menunjukkan fungsi f_i^R, masing-masing $\underline{p}_i$ menunjukkan predikat p_i^R dan masing-masing $\underline{c}_i$ menunjukkan konstan c_i^R.

Definisi 3.1.2.1

Bahasa pada tipe data R adalah himpunan yang beranggotakan string-string yang dibentuk dari anggota himpunan V, atau dapat dikatakan suatu bahasa adalah V^+. Anggota dari V disebut "*elemen*" dari bahasa.

contoh 3.1.2.1 :

Diberikan tipe data

$$N = \langle \omega, +^N, -^N, *^N, >^N, =^N, 0^N, 1^N, 2^N, \ldots \rangle$$

bila diberikan simbol khusus <u>if</u>, <u>then</u> dan <u>else</u>, maka dapat dibentuk

$$V = \mathrm{IVS} \cup \{\underline{+}, \underline{-}, \underline{*}, \underline{>}, \underline{=}, \underline{0}, \underline{1}, \underline{2}, \ldots\} \cup$$
$$\{\underline{(}, \underline{)}, \underline{,}\} \cup \{\underline{\text{if}}, \underline{\text{then}}, \underline{\text{else}}\}.$$

Sehingga bahasa pada tipe data N merupakan himpunan yang beranggotakan string-string yang dibentuk dari anggota himpunan V tersebut (V^+). Berikut ini diberikan contoh-contoh string dari bahasa pada tipe data N :

- 2+3
- (1+2)=3
- if x=4 then x-2 else 3*x

3.1.3 Bahasa Term

Sekarang akan dibahas masalah baru dengan tipe data :

$$R = \langle A, f_1^R, \ldots, f_n^R, P_1^R, \ldots, P_m^R, c_1^R, \ldots, c_r^R \rangle.$$

Akan didefinisikan suatu bahasa L, dan suatu fungsi $M : ENV \times L \longrightarrow A$ yang memetakan environment (environmet adalah fungsi $I : IVS \longrightarrow A$) dan suatu string dalam bahasa L ke nilai dari string dalam environment itu.

Definisi 3.1.2.1

Term	Fungsi M
i. jika c simbol konstan, maka c adalah term	$M(I,c) = c^R$
ii. jika v simbol variabel individu, maka v adalah term	$M(I,v) = I(v)$
iii. jika f simbol fungsi dengan n-parameter dan $t_1, \ldots, t_n$ term, maka $f(t_1, \ldots, t_n)$ adalah term	$M(I, f(t_1, \ldots, t_n)) = f^R(M(I,t_1), \ldots M(I,t_n))$

Bahasa L diatas disebut bahasa *Term*. Tanda punctuasi digarisbawahi untuk menekankan bahwa merupakan simbol formal dari elemen bahasa. Bahasa term yang telah didefinisikan diatas dapat dikatakan sebagai himpunan dari term.

Dari definisi diatas dapat dijelaskan sebagai berikut : Jika t simbol konstan $\underline{c}_k$, maka $M(I,t)$ adalah c_k^R, (jadi simbol konstan $\underline{c}_k$ menunjukkan obyek $c_k^R \in A$). Jika t simbol variabel individu $\underline{x}_i$, maka nilai dari $M(I,t)$ adalah $I(\underline{x}_i)$, nilai dari $\underline{x}_i$. Jika t adalah $\underline{f(t_1,\ldots,t_n)}$ dimana t_i adalah term, maka $M(I,t)$ adalah $f^R(M(I,t_1),\ldots M(I,t_n))$.

3.1.4 Bahasa Kondisi

Bahasa kondisi pada tipe data R merupakan perluasan sederhana dari bahasa term pada R. Akan diperkenalkan tiga simbol khusus yaitu <u>if</u>, <u>then</u> dan <u>else</u>. Bahasa kondisi dapat dikatakan sebagai himpunan kondisi seperti definisi berikut ini.

Definisi 3.1.4.1

Kondisi

---

i. setiap term adalah kondisi

ii. jika p adalah predikat dengan n-parameter dan $u_1,\ldots,u_n$, t_1,t_2 adalah kondisi maka

$$\underline{\text{if}}\ \underline{p(u_1,\ldots,u_n)}\ \underline{\text{then}}\ t_1\ \underline{\text{else}}\ t_2$$

adalah kondisi.

Sekarang didefinisikan fungsi evaluasi M untuk kondisi sebagai berikut :

Definisi 3.1.4.2

Fungsi M untuk kondisi
i. jika t term, maka $M(I,t) = M_{term}(I,t)$
ii-t. jika $p^R(M(I,u_1),\ldots,M(I,u_n)) = T$, maka
$M(I,\underline{if}\ p(u_1,\ldots,u_n)\ \underline{then}\ t_1\ \underline{else}\ t_2) = M(I,t_1)$
ii-f. jika $p^R(M(I,u_1),\ldots,M(I,u_n)) = F$, maka
$M(I,\underline{if}\ p(u_1,\ldots,u_n)\ \underline{then}\ t_1\ \underline{else}\ t_2) = M(I,t_2)$

Dari definisi diatas dapat dijelaskan sebagai berikut :

i. Jika t term, maka sesuai dengan definisi bahasa term $M(I,t) = M_{term}(I,t)$.

ii. Untuk menghitung $\underline{if}\ p(u_1,\ldots,u_n)\ \underline{then}\ t_1\ \underline{else}\ t_2$ dalam environment I, harus dihitung dulu $u_1,\ldots,u_n$ untuk mendapatkan $M(I,u_1),\ldots,M(I,u_n)$, kemudian ditentukan nilai dari $p^R(M(I,u_1),\ldots,M(I,u_n))$. Jika $p^R(M(I,u_1),\ldots,M(I,u_n))$ bernilai benar, maka nilai kondisinya adalah t_1, dan jika salah, maka nilai kondisinya adalah t_2.

3.2 BAHASA LOGIKA

3.2.1 Bahasa Logika Proposisional

Pada suatu directed graph $R \subseteq A \times A$ dengan $A=\{a,b,c\}$ dan $R = \{(a,b),(a,c),(b,a),(b,b),(b,c)\}$ dapat diperoleh beberapa kalimat pernyataan misalnya :

(i). (a,b) adalah anggota dari R.

(ii). (b,c) adalah anggota dari R.

(iii). (a,c) adalah bukan anggota dari R.

Kalimat-kalimat tersebut merupakan contoh dari *kalimat deklaratif* atau *proposisi* yaitu kalimat yang mengandung nilai benar atau salah. Untuk kalimat (i) bernilai benar, kalimat (ii) bernilai benar dan kalimat (iii) bernilai salah.

Logika proposisional adalah bahasan tentang bagaimana proposisi-proposisi dikombinasi ke bentuk proposisi-proposisi yang majemuk dengan suatu aturan. Karena setiap proposisi hanya mempunyai nilai T (benar) atau F (salah), maka logika proposisional akan mengolah tipe data yang hanya mempunyai nilai T dan F.

Didefinisikan tipe data B sebagai berikut :

$$B = \langle \{T,F\}, \&^B, V^B, \supset^B, \sim^B \rangle$$

dimana

$$\&^B : \{T,F\}^2 \longrightarrow \{T,F\}$$

$$V^B : \{T,F\}^2 \longrightarrow \{T,F\}$$

$$\supset^B : \{T,F\}^2 \longrightarrow \{T,F\}$$

$$\sim^B : \{T,F\} \longrightarrow \{T,F\}$$

yang didefinisikan dengan tabel seperti berikut :

x	y	$\&^B(x,y)$	$V^B(x,y)$	$\supset^B(x,y)$
F	F	F	F	T
F	T	F	T	T
T	F	F	T	F
T	T	T	T	T

x	$\sim^B(x)$
F	T
T	F

Tabel 3.2.1

Maksud dari fungsi-fungsi dalam tipe data B adalah sebagai berikut :

— $\&^B(x,y)$ menyatakan "x dan y".

$\&^B(x,y)$ = T jika dan hanya jika x dan y keduanya benar.

— $\vee^B(x,y)$ menyatakan "x atau y".

$\vee^B(x,y)$ = T bila hanya bila salah satu atau keduanya dari x,y benar.

— $\supset^B(x,y)$ menyatakan "jika x maka y".

Jika x salah maka untuk sembarang nilai y, $\supset^B(x,y)$ adalah benar, tetapi jika x benar maka $\supset^B(x,y)$ benar jika hanya jika y benar.

— $\sim^B(x)$ menyatakan "bukan x".

$\sim^B(x)$ benar jika dan hanya jika x salah.

Bahasa pada tipe data ini simbol variabel individunya dapat dikatakan sebagai *variabel proposisional* atau *simbol proposisi*. Akan digunakan $\underline{p},\underline{q},\underline{r}$, ... sebagai variabel proposisional.

Definisi 3.2.1.1

Suatu bentuk formula yang baik (*well formed formula* disingkat *wff*) didefinisikan sebagai berikut :

i. setiap simbol proposisi adalah wff

ii. jika α dan β adalah wff, maka $\underline{\&(\alpha,\beta)}$, $\underline{\vee(\alpha,\beta)}$, $\underline{\supset(\alpha,\beta)}$ dan $\underline{\sim(\alpha)}$ adalah wff.

Jika $\underline{p},\underline{q}$ dan $\underline{r}$ adalah simbol proposisi, maka $\underline{\&}(\underline{p},\underline{q})$ dan $\underline{\supset}(\underline{\&}(\underline{p},\underline{q}),\underline{\vee}(\underline{q},\underline{r}))$ merupakan wff. Agar lebih mudah dalam menuliskannya ditulis :

p&q daripada $\underline{\&(p,q)}$ dan

(p&q)⊃(q∨r) daripada $\underline{\supset}(\underline{\&}(\underline{p},\underline{q}),\underline{\vee}(\underline{q},\underline{r}))$.

Dengan penulisan baru ini simbol $\&, V, \supset$ dan $\sim$ dapat disebut *penghubung (logical connective)*. Sedangkan urutan operasi dari penghubung tersebut adalah :

$\sim$ dievaluasi pertama

$\&$

V

$\supset$ dievaluasi terakhir.

Dengan demikian

$pVq\&r$ sama dengan $pV(q\&r)$

$\sim p\supset qVr$ sama dengan $(\sim p)\supset(qVr)$

Operasi juga mulai dari kiri ke kanan, sehingga

$p\supset q\supset r$ sama dengan $(p\supset q)\supset r$

Definisi 3.2.1.2

Suatu wff adalah merupakan term dalam tipe data B.

Karena suatu wff merupakan term, maka dapat digunakan teknik - teknik pada bahasan bahasa term untuk mengevaluasi wff. Dalam hal ini environment I adalah fungsi $IVS \longrightarrow \{T,F\}$, dan dalam istilah logika environment dapat dikatakan sebagai <u>*interpretasi*</u> . Tabel 3.2.1.1. adalah *Tabel Kebenaran*, yaitu merupakan susunan tabel untuk mengevaluasi wff dalam semua environment yang mungkin. Pada masing-masing baris dari tabel kebenaran merupakan kombinasi nilai-nilai dari variabel-variabel dari wff. Sedangkan pada setiap kolom adalah wff yang akan dievaluasi.

p	q	pVq	~p	pVq⊃~p
F	F	F	T	T
F	T	T	T	T
T	F	T	F	F
T	T	T	F	F

p	~p	p&~p
F	T	F
T	F	F

p	q	r	p⊃q	q⊃r	(p⊃q)&(q⊃r)	p⊃r	(p⊃q)&(q⊃r)⊃(p⊃r)
F	F	F	T	T	T	T	T
F	F	T	T	T	T	T	T
F	T	F	T	F	F	T	T
F	T	T	T	T	T	T	T
T	F	F	F	T	F	F	T
T	F	T	F	T	F	T	T
T	T	F	T	F	F	F	T
T	T	T	T	T	T	T	T

Tabel 3.2.1.1 Tabel Kebenaran

Nilai dari M(I,t) tergantung dari I(v) untuk variabel-variabel v dalam t, sehingga jika wff t bernilai T pada setiap baris dari tabel kebenaran, maka wff t bernilai T dalam setiap interpretasi. Dengan demikian dari tabel kebenaran, terlihat bahwa (p⊃q)&(q⊃r)⊃(p⊃r) benar dalam setiap interpretasi.

Definisi 3.2.1.2

Suatu wff t adalah *tautologi* bila hanya bila untuk setiap I, M(I,t) = T, dan wff t adalah *kontradiksi* bila hanya bila untuk setiap I, M(I,t) = F.

Definisi 3.2.1.3

Dua wff t_1 dan t_2 adalah equivalen (t_1 eq t_2) bila hanya bila untuk setiap I, $M(I,t_1) = M(I,t_2)$.

Pada tabel 3.2.1.2. berikut menunjukkan pasangan dari wff yang equivalen, yang dapat mempunyai susunan berbeda tetapi mempunyai nilai yang sama.

Tabel 3.2.1.2. Beberapa pasangan wff yang equivalen

p&q	eq	p
pVp	eq	p
p&(q&r)	eq	(p&q)&r
pV(qVr)	eq	(pVq)Vr
pV(q&r)	eq	(pVq)&(pVr)
p&(qVr)	eq	(p&q)V(p&r)
~(pVq)	eq	~p&~q
~(p&q)	eq	~pV~q
~~p	eq	p
p⊃q	eq	~pVq

Theorema 3.2.1.1

Misalkan A dan B adalah wff, A eq B bila hanya bila (A⊃B)&(B⊃A) adalah tautologi.

<u>Bukti</u> :

⟹ Anggap A eq B. Dalam suatu environment I, ada dua kemungkinan yaitu M(I,A) = M(I,B) = T atau M(I,A) = M(I,B) = F.

Jika M(I,A) = M(I,B) = T, maka M(I,(A⊃B)&(B⊃A))=

$\&^B(\supset^B(T,T),\supset^B(T,T)) = \&^B(T,T) = T$.

Jika $M(I,A) = M(I,B) = F$, maka $M(I,(A\supset B)\&(B\supset A))=$ $\&^B(\supset^B(F,F),\supset^B(F,F)) = \&^B(T,T) = T$.

$\Leftarrow$ Anggap A dan B tidak equivalen. Maka ada environment I sedemikian hingga $M(I,A)\neq M(I,B)$. Bila $M(I,A)=T$ dan $M(I,B)= F$ maka $M(I,A\supset B)=F$ dan $M(I,(A\supset B)\&(B\supset A))=F$, sehingga $(A\supset B)\&(B\supset A)$ bukan tautologi. Jika $M(I,A)=F$ dan $M(I,B)=T$, maka dengan cara yang sama didapat $(A\supset B)\&(B\supset A)$ bukan tautologi.

3.2.2. Substitusi

Definisi 3.2.2.1

Substitusi adalah suatu fungsi σ dari himpunan variabel proposisional ke himpunan wff. Didefinisikan fungsi "Subst", yang membawa argumen wff dan substitusi dan memberikan hasil pengerjaan substitusi pada wff sebagai berikut :

$$\mathrm{Subst}(v,\sigma) = \sigma(v),\ \text{jika } v\in IVS$$

$$\mathrm{Subst}(\underline{\&}\llcorner t_1 \lrcorner t_2 \lrcorner,\sigma) = \underline{\&}\llcorner \mathrm{Subst}(t_1,\sigma)\lrcorner \mathrm{Subst}(t_2,\sigma)\lrcorner$$

$$\mathrm{Subst}(\underline{V}\llcorner t_1 \lrcorner t_2 \lrcorner,\sigma) = \underline{V}\llcorner \mathrm{Subst}(t_1,\sigma)\lrcorner \mathrm{Subst}(t_2,\sigma)\lrcorner$$

$$\mathrm{Subst}(\underline{\supset}\llcorner t_1 \lrcorner t_2 \lrcorner,\sigma) = \underline{\supset}\llcorner \mathrm{Subst}(t_1,\sigma)\lrcorner \mathrm{Subst}(t_2,\sigma)\lrcorner$$

$$\mathrm{Subst}(\underline{\sim}\llcorner t_1 \lrcorner,\sigma) = \underline{\sim}\llcorner \mathrm{Subst}(t_1,\sigma)\lrcorner$$

Theorema 3.2.2.1

Diberikan t adalah wff, σ suatu substitusi dan I environment. Jika didefinisikan environment I', dengan $I'(v)= M(I,\sigma(v))$ untuk $v\in IVS$, maka

$$M(I',t) = M(I,\mathrm{Subst}(t,\sigma))$$

<u>Bukti</u> :

Akan dibuktikan dengan induksi pada t.

*). Langkah dasar

Jika $v \in IVS$ maka

$M(I',v) = I'(v) = M(I,\sigma(v)) = M(I,Subst(v,\sigma))$

**). Langkah induksi

Jika $t = O \llcorner t_1 \lrcorner t_2 \lrcorner$ (dimana O mewakili $\underline{\&}$, $\underline{V}$ atau $\underline{\supset}$), maka

$$
\begin{aligned}
M(I',t) &= O^B(M(I',t_1),M(I',t_2)) && \text{(def. M)}\\
&= O^B(M(I,Subst(t_1,\sigma)),M(I,Subst(t_2,\sigma))) && \text{(HI)}\\
&= M(I,O\llcorner Subst(t_1,\sigma)\lrcorner Subst(t_2,\sigma)\lrcorner) && \text{(def. M)}\\
&= M(I,Subst(O\llcorner t_1 \lrcorner t_2 \lrcorner,\sigma)) && \text{(def. Subst.)}\\
&= M(I,Subst(t,\sigma))
\end{aligned}
$$

Kemudian jika $t = \underline{\sim}\llcorner t_1 \lrcorner$,

$$
\begin{aligned}
M(I',t) &= \sim^B(M(I',t_1)\\
&= \sim^B(M(I,Subst(t_1,t))\\
&= M(I,\underline{\sim}\llcorner Subst(t_1,\sigma)\lrcorner)\\
&= M(I,Subst(\underline{\sim}\llcorner t_1 \lrcorner),\sigma)\\
&= M(I,Subst(t,\sigma))
\end{aligned}
$$

Theorema 3.2.2.2 (Theorema Substitusi)

Jika t_1 eq t_2, maka $Subst(t_1,\sigma)$ eq $Subst(t_2,\sigma)$

<u>Bukti</u> :

Anggap t_1 eq t_2 dan I suatu environmet. Jika diberikan I', dengan $I' = M(I,\sigma(v))$, maka dengan theorema 3.2.2.1. didapat :

$$M(I,Subst(t_1,\sigma)) = M(I',t_1) = M(I',t_2) = M(I,Subst(t_2,\sigma))$$

sehingga dengan definisi 3.2.1.3. diperoleh $Subst(t_1,\sigma)$ equivalen dengan $Subst(t_2,\sigma)$

Theorema 3.2.2.3

Jika σ_1 dan σ_2 adalah substitusi sedemikian sehingga untuk suatu v∈IVS $\sigma_1(v)$ eq $\sigma_2(v)$, maka untuk t suatu wff,

$$\mathrm{Subst}(t,\sigma_1) \text{ eq } \mathrm{Subst}(t,\sigma_2)$$

Bukti :

Diberikan I dan I′ environment dimana $I'(v)=M(I,\sigma_1(v))$. Karena $\sigma_1(v)$ eq $\sigma_2(v)$, maka untuk v∈IVS $I'(v)=M(I,\sigma_2(v))$. Sehingga dengan theorema 3.2.2.1. dan theorema 3.2.2.2. didapat

$$M(I,\mathrm{Subst}(t,\sigma_1)) = M(I',t) = M(I,\mathrm{Subst}(t,\sigma_2)).$$

3.2.3 Bahasa Logika Tingkat Satu

Diberikan tipe data

$$R = \langle A, f_1^R, \ldots, f_n^R, p_1^R, \ldots, p_m^R, c_1^R, \ldots, c_r^R \rangle$$

Akan didefinisikan bahasa dari "formula tingkat satu" dalam R, dan himpunan dari formula-formula secara induktif bersama dengan definisi induktif dari fungsi M. Dalam kasus ini fungsi M akan memetakan suatu environment dan suatu formula ke nilai T atau F.

Definisi 3.2.3.1

Himpunan dari *formula tingkat satu* dalam R adalah himpunan string-string yang didefinisikan sebagai berikut :

1. jika p adalah simbol predikat dengan n-parameter dan $t_1,\ldots,t_n$ term, maka $p(t_1,\ldots,t_n)$ formula

2.1. jika G dan H formula, maka &GH adalah formula

2.2. jika G dan H formula, maka $\underline{\vee}GH$ adalah formula

2.3. jika G dan H formula, maka $\underline{\supset}GH$ adalah formula

2.4. jika G formula, maka $\underline{\sim}G$ adalah formula

3.1. jika G formula, dan v simbol variabel individu, maka $\underline{\exists}vG$ adalah formula

3.2. jika G formula, dan v simbol variabel individu, maka $\underline{\forall}vG$ adalah formula

Pada tabel berikut memberikan nama-nama dari definisi diatas.

Formula	Nama
1.	Formula atomic
2.1.	Konjungsi
2.2.	Disjungsi
2.3.	Implikasi
2.4.	Negasi
3.1.	Existensial
3.2.	Universal

Tabel 3.2.3.1

Sekarang akan didefinisikan fungsi evaluasi M untuk formula-formula tingkat satu. Jika G suatu formula dan I environment maka fungsi M dapat digunakan untuk menjawab apakah formola G benar dalam interpretasi I.

Pada definisi yang pertama $p(t_1, \ldots, t_n)$ disebut formula atomic. Masing-masing term t_i berhubungan dengan suatu elemen himpunan A dari nilai-nilai dalam tipe data, yaitu $M(I,t_i)$. Simbol predikat p menunjukkan predikat p^R : $A^n \longrightarrow \{T,F\}$. Sehingga dapat dikatakan bahwa $p(t_1,\ldots,t_n)$ adalah pertanyaan "Apakah p^R benar $M(I,t_1),\ldots,M(I,t_n)$?". Oleh karena itu dapat ditulis :

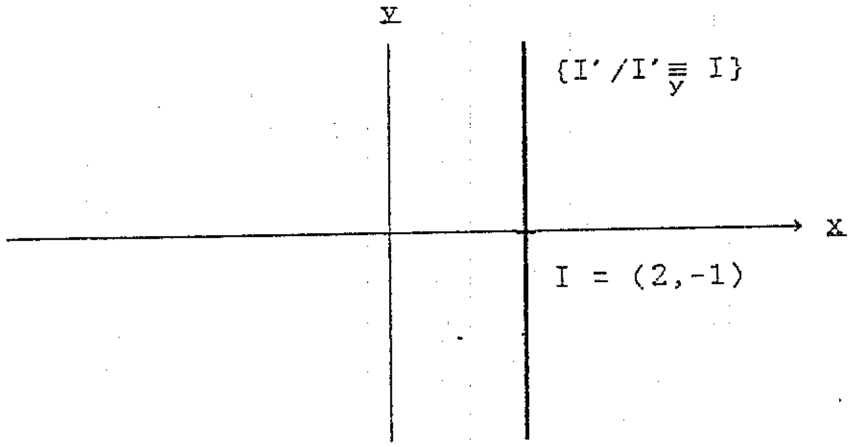
$$M(I,p(t_1,\ldots,t_n)) = p^R(M(I,t_1),\ldots,M(I,t_n))$$

Formula atomic menyatakan suatu masalah dalam tipe data tentang benar tidaknya masalah tersebut, dan bukan merupakan hasil kombinasi dari beberapa proposisi. Dengan demikian formula atomic merupakan proposisi tunggal dari tipe data. Untuk formula 2.1. sampai dengan 2.4. merupakan kombinasi dari proposisi seperti pada logika proposisional. Sedangkan pada formula 3.1. dan 3.2., dalam operasinya dinamakan *quantifier*. Sebelum mendefinisikan fungsi M untuk formula 3.1 dan 3.2 dibutuhkan definisi bantu sebagai berikut :

Definisi 3.2.3.2

Bila I dan I' adalah interpretasi, dan v simbol variabel individu, maka I dan I' adalah equivalen modulo v (ditulis $I \underset{v}{\equiv} I'$) bila hanya bila untuk setiap simbol variabel individu v, jika $w \neq v$, maka $I(w) = I'(w)$.

Dengan kata lain, I dan I' equivalen modulo v, bila keduanya selalu sama kecuali mungkin pada v. Sebagai contoh, pada bilangan-bilangan riil diberikan dua variabel $\underline{x}$ dan $\underline{y}$, dengan $I(\underline{x})=2$ dan $I(\underline{y})=-1$. Maka $\{I'/I' \underset{y}{\equiv} I\}$ dapat ditunjukkan dengan gambar berikut :



Gambar 3.2.3.1

Sekarang fungsi M untuk formula tingkat satu dapat didefinisikan pada tabel berikut :

1.	$M(I,\underline{p}(t_1,\ldots,t_n)) = p^R(M(I,t_1),\ldots,M(I,t_n))$
2.1.	$M(I,\underline{\&}GH) = \&^B(M(I,G),M(I,H))$
2.2.	$M(I,\underline{\vee}GH) = \vee^B(M(I,G),M(I,H))$
2.3.	$M(I,\underline{\supset}GH) = \supset^B(M(I,G),M(I,H))$
2.4.	$M(I,\underline{\sim}G) = \sim^B(M(I,G))$
3.1.	$M(I,\underline{\exists}vG)$ = T bila hanya bila untuk suatu I', I' $\underset{v}{\equiv}$ I dan M(I',G)=T
3.2.	$M(I,\underline{\forall}vG)$ = T bila hanya bila untuk setiap I' sedemikian hingga I' $\underset{v}{\equiv}$ I, M(I',G)=T

Tabel 2.4.3.2. Tabel fungsi M untuk formula tingkat satu

Pada definisi (3.1) $\underline{\exists}$vG dibaca "ada v sedemikian sehingga G". Dengan kata lain $\underline{\exists}$vG benar dalam interpretasi I bila hanya bila dapat dipilih suatu nilai untuk v yang membuat G benar. Karena tidak ada suatu ketetapan untuk variabel-variabel yang lain dalam G, maka harus didapatkan rumusan secara tetap. Dari kedua kalimat diatas dapat diambil kesimpulan bahwa $\underline{\exists}$vG menjadi benar bila hanya bila ada suatu I' yang berbeda dengan I hanya pada v, sedemikian hingga M(I',G)=T.

Pada definisi (3.2), ∀vG dibaca "untuk semua v, maka G",yaitu $\underline{\forall}$vG manjadi benar pada I bila hanya bila untuk setiap nilai dari v membuat G benar. Pada semua variabel yang lain interpretasi-interpretasi I' sama dengan I kecuali mungkin pada v. Sehingga dapat dikatakan bahwa $\underline{\forall}$vG benar pada I bila hanya bila G benar pada semua I'.

Bila I($\underline{x}$)=2, I($\underline{y}$)=-1 dan G formula, maka dengan melihat gambar 3.2.3.1. tampak bahwa $\underline{\exists}$yG benar pada I jika hanya jika G benar pada suatu titik dari I' pada garis vertical. $\underline{\forall}$vG benar pada I bila hanya bila G benar pada setiap titik dari I' pada garis vertical.

3.2.4 Tingkat Kebenaran

Pada bahasan yang lalu, telah dibahas masalah penentuan kebenaran atau kesalahan dari formula dalam tipe data dalam suatu interpretasi. Pada bahasan ini akan dibahas bagaimana suatu formula dikatakan *valid* dalam suatu tipe data.

Definisi 3.2.4.1

Diberikan R suatu tipe data, dikatakan R mempunyai predikat persamaan bila, R mempunyai predikat $=^R$ yang didefinisikan sebagai berikut :

$$=^R(x,y) = \begin{cases} T & \text{jika } x = y \\ F & \text{jika } x \neq y \end{cases}$$

Suatu tipe data dapat dikonversi menjadi tipe data yang mempunyai predikat persamaan dengan menambahkan predikat $=^R$. Sehingga dapat dikatakan semua tipe data mempunyai predikat persamaan.

Definisi 3.2.4.2

Diberikan G suatu formula dan R suatu tipe data dengan predikat persamaan, dikatakan G valid dalam R (ditulis $R \models G$) bila hanya bila untuk setiap fungsi $I:IVS \longrightarrow A$, $M(I,G)=T$.

G valid dalam R bila hanya bila G benar dalam setiap environment pada R. Notasi $R \models G$ mempunyai arti "R membuat G benar", "R model untuk G" atau "R memenuhi G".

contoh 3.2.4.1.

Diberikan tipe data $Z = \langle \mathcal{Z},+,-,*,<,=,0,1\rangle$ dengan $\mathcal{Z}$ adalah himpunan bulat, maka formula-formula berikut valid dalam Z :

i. $(\forall x)(\forall y)[x+y = y+x]$

ii. $(\forall x)(\forall y)(\forall z)[(x+y)+z = x+(y+z)]$

iii. $(\forall x)(\forall y)(\forall z)[x<y \& y<z \supset x<z]$

Definisi 3.2.4.3

Jika G formula, maka G valid jika hanya jika untuk setiap tipe data R dengan predikat persamaan, $R \models G$.

Jadi G valid bila hanya bila G valid dalam setiap tipe data.

contoh 3.2.4.2

i. $(\forall x)(\forall y)[x=y \supset y=x]$

ii. $(\exists x)[x=x]$

3.2.5 Provabilitas

Pada umumnya tidak mungkin menentukan kevalidan suatu formula secara langsung dari definisi kevalidan. Sekarang akan diberikan suatu definsi induktif langsung dari himpunan wff yang valid.

Suatu definisi induktif himpunan *S*, yang merupakan suatu himpunan dasar dari formula-formula dan beberapa aturan (rule) , adalah "jika $x \in S$ maka $f(x) \in S$", dengan x adalah suatu formula dan f suatu aturan (rule). Dalam logika, formula-formula dalam himpunan dasar tersebut disebut *axioma*, aturan-aturannya disebut *aturan kesimpulan (rules of inference)*. Kemudian suatu penurunan yang menunjukkan bahwa suatu formula merupakan anggota dari *S* disebut *bukti formal (formal proof)*, dan suatu formula yang terdapat dalam himpunan tersebut disebut *provable* (dapat dibuktikan). Definisi dalam bentuk ini disebut *sistim formal*.

Diberikan FW adalah himpunan dari semua himpunan berhingga wff, dan didefinisikan suatu relasi $\models$ pada FW sebagai berikut :

Definisi 3.2.5.1

Jika $\Delta \in FW$ dan $\Gamma \in FW$, maka $(\Delta,\Gamma) \in \models$ bila hanya bila setiap interpretasi yang membuat semua formula dalam Δ benar, membuat sekurang-kurangnya satu formula dalam Γ benar.

Sebagai contoh bila I(p)=F, I(q)=T, dan I(r)=T, maka

$$(\{p, p \supset q \vee \sim r\}, \{q, \sim r\}) \in \models$$

tetapi

$$(\{p \vee q, \sim q \supset r\}, \{\sim p \supset \sim r\}) \notin \models$$

karena I membuat $p \vee q$ dan $\sim q \supset r$ benar, dan $\sim p \supset \sim r$ salah.

Notasi $(\Delta,\Gamma) \in \models$ dapat diganti dengan $\Delta \models \Gamma$, dan semua tanda kurung dapat dihilangkan sehingga dapat ditulis

$$p, p \supset q \vee \sim r \models q, \sim r$$

dengan urutan operasinya, $\models$ adalah paling akhir.

Dalam hal ini $\models$ bukan merupakan definisi induktif yang akan ditelusuri, karena masih mencakup interpretasi-interpretasi. Sekarang akan didefinisikan suatu relasi $\vdash$ pada FW, dan akan dibuktikan dengan teori fundamental induksi bahwa $\vdash \subseteq \models$.

Agar lebih mudah dibaca akan digunakan notasi $\Delta \vdash \Gamma$ dan menghilangkan semua tanda kurung seperti pada pengerjaan $\models$. Selain itu ditulis Δ, A sebagai $\Delta \cup \{A\}$. Akan ditulis

$$\frac{\Delta \vdash \Gamma}{\Delta' \vdash \Gamma'}$$

untuk "jika $(\Delta,\Gamma) \in \vdash$ maka $(\Delta',\Gamma') \in \vdash$ " dan

$$\frac{\Delta_1 \vdash \Gamma_1 \quad \Delta_2 \vdash \Gamma_2}{\Delta_3 \vdash \Gamma_3}$$

untuk "jika $(\Delta_1,\Gamma_1) \in \vdash$ dan $(\Delta_2,\Gamma_2) \in \vdash$ maka $(\Delta_3,\Gamma_3) \in \vdash$ ", sedangkan untuk memberi nama aturan dituliskan disebelah kanan aturan tersebut ssperti dibawah ini

$$\frac{\Delta_1 \vdash \Gamma_1 \quad \Delta_2 \vdash \Gamma_2}{\Delta_3 \vdash \Gamma_3} \quad \xi$$

tanda ξ menunjukkan nama.

Tabel dibawah ini menunjukkan definisi induktif dari $\vdash$, yang disebut definisi *sistim G*.

(1)	Jika $\Delta \cap \Gamma \neq \emptyset$, maka $\Delta \vdash \Gamma$	(Axioma)	
(2)	$\frac{\Delta, A \vdash \Gamma \quad \Delta, B \vdash \Gamma}{\Delta, A \vee B \vdash \Gamma}$ $\vee\vdash$	$\frac{\Delta \vdash A, B, \Gamma}{\Delta \vdash A \vee B, \Gamma}$ $\vdash\vee$	ATURAN
(3)	$\frac{\Delta, A, B \vdash \Gamma}{\Delta, A \& B \vdash \Gamma}$ $\&\vdash$	$\frac{\Delta \vdash A, \Gamma \quad \Delta \vdash B, \Gamma}{\Delta \vdash A \& B, \Gamma}$ $\vdash\&$	
(4)	$\frac{\Delta \vdash \Gamma, A \quad A, B \vdash \Gamma}{\Delta, A \supset B \vdash \Gamma}$ $\supset\vdash$	$\frac{\Delta, A \vdash B, \Gamma}{\Delta \vdash A \supset B, \Gamma}$ $\vdash\supset$	
(5)	$\frac{\Delta \vdash A, \Gamma}{\Delta, \sim A \vdash \Gamma}$ $\sim\vdash$	$\frac{\Delta, A \vdash \Gamma}{\Delta \vdash \sim A, \Gamma}$ $\vdash\sim$	
	(a)	(b)	

Tabel 3.2.5.1 Sistim G

Bagaimana menggunakan sistim diatas ? Diberikan pasangan (Δ,Γ) dan akan ditentukan apakah pasangan tersebut merupakan anggota dari $\vdash$. Caranya adalah dengan memilih aturan dari definisi diatas yang cocok yang memasukkan (Δ,Γ), sehingga membentuk

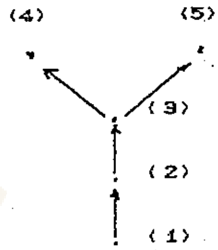
$$\frac{\Delta' \vdash \Gamma'}{\Delta \vdash \Gamma}$$

Kemudian dengan cara yang sama mengerjakan (Δ',Γ'), sampai pada akhirnya mencapai suatu axioma. Dalam hal ini jalan yang terbaik adalah mengerjakan dengan tree yang dimulai dari (Δ,Γ) sebagai root, kemudian maju ke cabang-cabang sampai mencapai axioma-axioma atau suatu jalan buntu. Jika di setiap ujung cabang membetuk axioma maka $(\Delta,\Gamma)\in\vdash$ atau dikatakan $A\vdash\Gamma$ adalah anggota dari sistim G. Jika pada suatu ujung cabang menemui jalan buntu maka $(\Delta,\Gamma)\notin\vdash$ atau bukan anggota dari G.

contoh 3.2.5.1

Akan diuji apakah $(\emptyset,\{((p\supset q)\&p)\supset q\}) \in \vdash$.

$$\dfrac{(2)\;\dfrac{(3)\;\dfrac{\overset{(4)}{p\vdash p,q} \qquad \overset{(5)}{p,q\vdash q}}{p\supset q, p\vdash q}\supset\vdash}{(p\supset q)\&p\vdash q}\&\vdash}{(1)\;\vdash ((p\supset q)\&p)\supset q}\vdash\supset$$



Dengan menggunakan aturan $\vdash\supset$ (4b) maka $(\emptyset,\{((p\supset q)\&p)\supset q\})$ (1) dengan $\Delta=\Gamma=\emptyset$, $A=(p\supset q)\&p$ dan $B=q$, akan menghasilkan (2) yaitu $(\{(p\supset q)\&p\},\{q\})$. Kemudian dengan dengan menggunakan aturan $\&\vdash$ (3b) dari (2) akan menghasilkan (3) yaitu $(\{p\supset q,p\},\{q\})$. Selanjutnya dengan aturan $\supset\vdash$ (4a) dari (3) menghasilkan (4) dan (5) yaitu $(\{p\},\{p,q\})$ dan $(\{p,q\},\{q\})$. Karena masing-masing pasangan himpunan (4) dan (5) mempunyai elemen yang sama, maka (4) dan (5) merupakan axioma. Dengan demikian $(\emptyset,\{((p\supset q)\&p)\supset q\})\in\vdash$.

contoh 3.2.5.2

Akan diuji apakah $(pVq)\&\sim p\vdash q$ anggota dari G.

Dengan menyusun tree dengan $(pVq)\&\sim p\vdash q$ sebagai root diperoreh susunan (a) atau (b) sebagai berikut :

$$\dfrac{\dfrac{\dfrac{p \vdash p,q}{p,\ p \vdash q}\ {\sim}\vdash \qquad q,{\sim}p \vdash q}{pVq),{\sim}p \vdash q}\ V\vdash}{(pVq)\&{\sim}p \vdash q}\ \&\vdash$$

(a)

$$\dfrac{\dfrac{\dfrac{p \vdash p,q \qquad q \vdash p,q}{pVq \vdash p,q}\ V\vdash}{pVq,\ p \vdash q}\ {\sim}\vdash}{(pVq)\&{\sim}p \vdash q}\ \&\vdash$$

(b)

Walaupun susunan dari tree (a) dan (b) berbeda, tapi dari kedua susunan tersebut menunjukkan bahwa (pVq)&~p⊢q adalah anggota dari G.

Dari contoh 3.2.5.2 dapat diambil kesimpulan bahwa bentuk tree boleh sembarang asalkan memenuhi aturan-atuaran dalam sistim G. Kemudian akan diberi suatu contoh dalam penyusunan tree yang menemui jalan buntu.

contoh 3.2.5.3

Akan diuji apakah p⊃q,p,r⊢r&~q anggota dari G. Disusun tree sebagai berikut.

$$\dfrac{p\supset q,p,r \vdash r \qquad \dfrac{\dfrac{p,r,q \vdash p \qquad \overset{?\ ?\quad ?\ \ ?}{q,p,r \vdash}}{p\supset q,p,r,q \vdash}\ \supset\vdash}{p\supset q,p,r \vdash {\sim}q}\ \vdash{\sim}}{p\supset q,p,r \vdash r\&{\sim}q}\ \vdash\&$$

Karena $q,p,r\vdash$ bukan merupakan axioma dan tidak ada kelanjutannya, maka $p\supset q,p,r\vdash r\&\sim q$ bukan anggota dari G.

Dari ketiga contoh tersebut dapat diambil suatu kesimpulan bahwa jika $(\Delta,\Gamma)\in\vdash$, maka dapat disusun tree sembarang, sehingga pada setiap ujung cabang merupakan axioma.

Theorema 3.2.5.1

Jika $\Delta\vdash\Gamma$ maka $\Delta\models\Gamma$

Bukti :

Akan ditunjukkan bahwa jika $(\Delta.\Gamma)$, adalah axioma dari sistim G, maka $\Delta\models\Gamma$. Selain itu $\models$ adalah tertutup dalam aturan-aturan dari sistim G dan dengan theorema fundamental induksi didapat $\vdash\subseteq\models$. Akan dibuktikan dengan induksi yang dimulai dari langkah dasar dan satu langkah untuk masing aturan.

*). Langkah dasar

Diberikan $\Delta\vdash\Gamma$ suatu axioma dan $C\in\Delta\cap\Gamma$. Setiap interpretasi I yang membuat semua wff dalam Δ benar, membuat C benar, dengan demikian I membuat satu wff dalam Γ benar yaitu C.

**). Langkah induksi

2a. Aturan $\vee\vdash$: Anggap $\Delta,A\models\Gamma$ dan $\Delta,B\models\Gamma$. Bila interpretasi I membuat semua wff dalam $\Delta,A\vee B$ benar maka I membuat A benar atau B benar. Bila I membuat A benar, maka I membuat semua wff dalam Δ,A benar dan dengan demikian I membuat beberapa formula dalam Γ benar. Jika

I membuat B benar, maka I membuat semua wff dalam Δ,B benar dan dengan demikian I membuat beberapa formula dalam Γ benar. Oleh karena I membuat beberapa wff dalam Γ benar, maka Δ,AVB$\models\Gamma$.

2b. Aturan $\vdash$V : Anggap $\Delta\models$ A,B,Γ. Akan ditunjukkan bahwa $\Delta\models$ AVB,Γ. Misal interpretasi I membuat semua wff dalam Δ benar. Karena $\Delta\models$ A,B,Γ maka I membuat beberapa wff dalam A,B,Γ benar. Jadi dapat dikatakan I membuat formula A benar atau B benar atau beberapa wff dalam Γ benar. Jadi $\Delta\models$ AVB,Γ.

3a. Aturan &$\vdash$: Anggap Δ,A,B$\models$ Γ. Bila interpretasi I membuat semua formula dalam $\Delta\cup$\{A&B\} benar, maka I membuat A dab B benar. Oleh karena itu I membuat semua formula dalam $\Delta\cup$\{A,B\} benar. Karena Δ,A,B$\models$ Γ, maka interpretasi I membuat membuat beberap formula dalam Γ benar. Sehingga Δ,A&B$\models\Gamma$.

3b. Aturan $\vdash$& : Anggap $\Delta\models$ A,Γ dan $\Delta\models$ B,Γ. Akan ditunjukkan bahwa $\Delta\models$ A&B,Γ. Interpretasi I membuat semua formula dalam Δ benar. Harus ditunjukkan I membuat beberapa formula dalam Γ benar atau I membuat A&B benar. Karena $\Delta\models$ A,Γ dan $\Delta\models$B,Γ, maka I membuat beberapa formula dalam Γ benar atau I membuat kedua A dan B benar. Sehingga I membuat beberapa formula dalam $\Gamma\cup$\{A&B\} benar.

4a. Aturan $\supset\vdash$: Anggap $\Delta\models\Gamma$,A dan Δ,B$\models\Gamma$. Bila interpretasi I membuat semua formula dalam $\Delta, A\supset B$ benar, maka I membuat semua formula dalam Δ benar dan membuat $A\supset B$ benar. Oleh karena itu pasti I membuat B benar dan dengan demikian membuat semua formula dalam Δ, A benar, sehingga I membuat beberapa formula dalam Γ benar. I membuat $A\supset B$ benar bila hanya bila I membuat B benar dan tidak harus membuat A benar, sehingga sudah barang tentu I membuat beberapa formula dalam Γ, A benar. Jadi $\Delta, A\supset B \models \Gamma$.

4b. Aturan $\vdash\supset$: Anggap $\Delta, A \models B, \Gamma$. Akan ditunjukkan bahwa $\Delta \models A\supset B, \Gamma$. Misal I membuat semua formula dalam Δ benar. Karena $\Delta, A \models B, \Gamma$ maka I membuat semua formula dalam Δ, A benar dan membuat beberapa formula dalam B, Γ benar. Oleh karena itu I membuat A benar, sehingga I membuat $A\supset B$ benar bila hanya bila I membuat B benar. Karena I membuat beberapa formula dalam Γ, B benar maka I membuat beberapa formula dalam $A\supset B, \Gamma$ benar.

5a. Aturan $\sim\vdash$: Anggap $\Delta \models A, \Gamma$. Misal interpretasi I membuat semua formula dalam $\Delta, \sim A$ benar. Oleh karena itu I membuat semua formula dalam Δ benar dan I membuat A salah. Karena $\Delta \models A, \Gamma$ maka I membuat beberapa formula dalam A, Γ benar. Karena I membuat A salah maka I harus membuat beberapa wff dalam Γ benar. Sehingga $\Delta, \sim A \models \Gamma$.

5b. Anggap $\Delta, A \models \Gamma$. Akan ditunjukkan bahwa $\Delta \models \sim A, \Gamma$. Diberikan interpretasi I membuat semua formula dalam Δ benar. Karena $\Delta, A \models \Gamma$, maka I membuat A benar, dan kemudian membuat beberapa formula dalam Γ benar. Sehingga I membuat $\sim A$ salah. Karena I membuat beberapa formula dalam Γ benar, maka membuat beberapa formula dalam $\{\sim A\}\cup\Gamma$ benar. Sehingga $\Delta\models \sim A,\Gamma$.

3.3 BAHASA STATEMEN

Dalam bahasan ini akan didefinisikan bahasa dari statemen-statemen, yang dalam opersi dasarnya merupakan assignment dari suatu nilai ke variabel.

Definisi 3.3.1

Jika R adalah tipe data, himpunan dari statemen-statemen adalah himpunan dari string-string yang didefinisikan sebagai berikut :

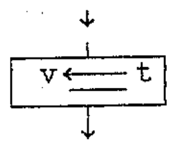
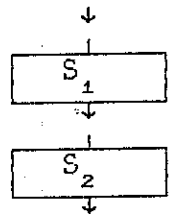
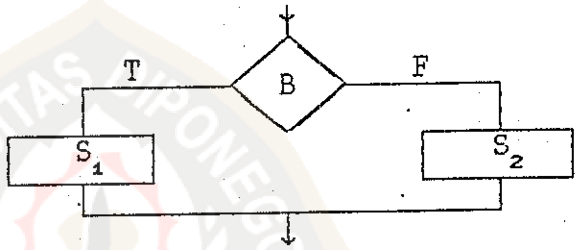
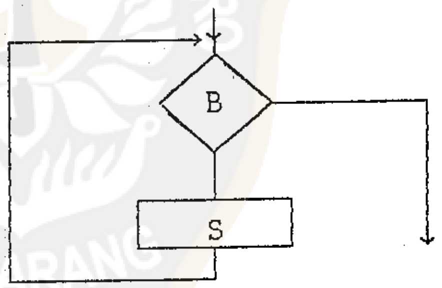
i. jika $v \in IVS$ dan t adalah term dalam R, maka $v \Longleftarrow t$ adalah statemen

ii. jika S_1 dan S_2 statemen maka <u>begin</u> $S_1;S_2$ <u>end</u> adalah statemen

iii. jika B formula dalam R yang tanpa quantifier dan S_1,S_2 statemen, maka <u>if</u> B <u>then</u> S_1 <u>else</u> S_2 adalah statemen

iv. jika B formula dalam R yang tanpa quantifier dan S_1 statemen, maka <u>while</u> B <u>do</u> S adalah statemen

Dari definisi diatas statemen-statemennya dapat digambarkan sebagai flowchart berikut :

$v \Longleftarrow t$ statemen assignmen	
<u>begin</u> $S_1 ; S_2$ <u>end</u> statemen campuran	
<u>if</u> B <u>then</u> S_1 <u>else</u> S_2 statemen kondisi	
<u>while</u> B <u>do</u> S statemen <u>while</u>	

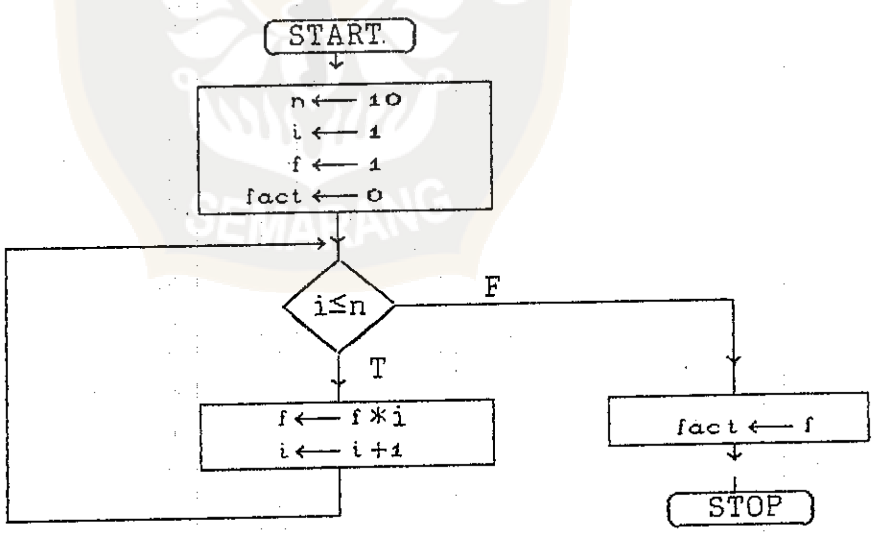
Gambar 3.3.1

Untuk definisi (ii) dapat dikembangkan, jika $S_1,\ldots,S_n$ adalah statemen, maka <u>begin</u> $S_1,\ldots,S_n$ <u>end</u> adalah statemen. Kombinasi dari statemen-statemen yang sesuai dengan bahasa statemen tersebut dapat menghasilkan susunan perintah atau kerja yang disebut *program.* Dibawah ini diberikan contoh program dan gambar flowchartnya.

Contoh 3.3.1

Program faktorial dari bilangan bulat positif 10 (10!)

```
begin
    n← 10;
    i← 1;
    f← 1;
    fact← 0;
    while i≤n do
        begin
            f← f*i;
            i← i+1
        end;
    fact← f
end
```



Gambar 3.3.1 Flowchart

Sekarang akan didefinisikan fungsi evaluasi M. Operasi statemen-statemen dengan perubahan nilai-nilai dari variabel-variabel, yang konsepnya berupa pemetaan dari nama-nama variabel ke nilai-nilainya adalah environment. Dengan demikian akan dibuat fungsi M yang membawa statemen S dan environment I(environment awal) ke environment I'-(environment seteleh S dijalankan). Tabel berikut adalah tabel definisi fungsi M.

i.	$M(I, v \Longleftarrow t) = I'$, dimana $I' = \begin{cases} M_{term}(I,t) & \text{jika } w \neq v \\ I(w) & \text{jika } w = v \end{cases}$
ii.	$M(I,$ <u>begin</u> $S_1 ; S_2$ <u>end</u>$) = M(M(I,S_1),S_2)$
iii-t	jika $M_f(I,B)=T$, maka $M(I,$<u>if</u> B <u>then</u> S_1 <u>else</u> $S_2) = M(I,S_1)$
iii-f	jika $M_f(I,B)=F$, maka $M(I,$<u>if</u> B <u>then</u> S_1 <u>else</u> $S_2) = M(I,S_2)$
iv-f	jika $M_f(I,B)=F$, maka $M(I,$<u>while</u> B <u>do</u> S$) = I$
iv-t	jika $M_f(I,B)=T$, maka $M(I,$<u>while</u> B <u>do</u> S$) = M(M(I,S),$ <u>while</u> B <u>do</u> S$)$

Tabel 3.3.1 Definisi fungsi M untuk statemen

Pada tabel diatas M_{term} pada nomor (i) menunjukkan fungsi evaluasi untuk term dan M_f pada (iii) - (iv) menunjukan fungsi evaluasi untuk formula. Dalam bentuk fungsional fungsi M ditulis $M(I,S)=I'$, sedangkan dalam bentuk relasional ditulis $(I,S,I') \in M$. Bila untuk I,S, tidak terdapat I' sedemikian hingga $(I,S,I') \in M$ atau $M(I,S)$ tak terdefinisi maka program tidak pernah berhenti.