

BAB II

TEORI PENUNJANG

Teori-teori penunjang yang akan digunakan dalam penulisan tugas akhir ini meliputi proposisi dan predikat kalkulus, himpunan, relasi dan fungsi serta spesifikasi perangkat lunak.

2.1. PROPOSISI DAN PREDIKAT KALKULUS

Proposisi dan predikat kalkulus merupakan bahasa matematik yang digunakan untuk mengekspresikan pernyataan-pernyataan logika.

2.1.1. Proposisi kalkulus

2.1.1.1. Aturan Sintaks Bahasa Logika Proposisi

Definisi 2.1 :

Kalimat logika proposisi dibentuk oleh simbol-simbol kebenaran dan simbol-simbol proposisi. Simbol kebenaran berupa nilai *benar* atau *salah*, sedangkan simbol proposisi dapat berupa abjad (P, Q, R) ataupun berupa variabel proposisional seperti *nm_barang*, *daftar_mhs* dan sebagainya.

Definisi 2.2 :

Kalimat logika proposisi dibentuk dengan menggunakan penghubung proposisi, yaitu :

1. negasi (\neg) dengan arti *bukan*
2. konjungsi (\wedge) dengan arti *dan*
3. disjungsi (\vee) dengan arti *atau*
4. implikasi (\Rightarrow) dengan arti *jika.. maka..*
5. biimplikasi (\Leftrightarrow) dengan arti *jika dan hanya jika*
6. eksor (\vee_e) dengan arti *bukan (jika dan hanya jika)*

Keterangan :

Untuk nomor 1 merupakan operator dasar terhadap satu proposisi.

Untuk nomor 2-6 merupakan operator dasar terhadap dua proposisi.

Contoh 2.1:

$$\text{nm_barang} \wedge \text{daftar_mhs} \Rightarrow \text{transaksi}$$

Kalimat di atas menyatakan jika barang ada dan nama mahasiswa tercantum pada daftar mahasiswa, maka transaksi dilaksanakan.

Definisi 2.3 :

Dapat dibentuk kalimat yang lebih kompleks yang memiliki kalimat-kalimat pembentuk atau anak kalimat. Dengan demikian untuk kalimat kompleks E maka E sendiri juga merupakan anak kalimat dari E dan disebut anak kalimat sejati.

Contoh 2.2 :

$$E : [(\neg (P \vee Q)) \Leftrightarrow ((\neg P) \wedge (\neg Q))]$$

Kalimat diatas memiliki anak-anak kalimat :

- | | |
|------------------------|-----------------------------------|
| 1. P | 5. $\neg P$ |
| 2. Q | 6. $\neg Q$ |
| 3. $(P \vee Q)$ | 7. $(\neg P) \wedge (\neg Q)$ |
| 4. $\neg (P \vee Q)$ | 8. E |

2.1.1.2. Aturan Semantik Bahasa Logika Proposisi

Nilai kebenaran kalimat logika proposisi ditentukan oleh informasi-informasi tentang kebenaran dari proposisi-proposisi yang membentuk kalimat tersebut.

Definisi 2.4:

Kebenaran-kebenaran dasar operator dan penghubung logika proposisi :

1. Jika P benar maka $(\neg P)$ salah
2. P atau Q benar jika salah satu P atau Q benar
3. P dan Q benar jika P maupun Q keduanya benar
4. $P \Rightarrow Q$ bernilai salah jika P benar dan Q salah
5. P jika hanya jika Q bernilai benar, jika P dan Q bernilai sama keduanya
6. P eksor Q bernilai salah, jika P dan Q bernilai sama

Informasi- informasi yang sesuai dengan bahasan pada definisi 2.4 untuk selanjutnya disebut *interpretasi* dan diberikan notasi dasar I.

Definisi 2.5 :

I merupakan suatu interpretasi kalimat F jika I memberi nilai kebenaran pada setiap simbol yang ada di F. I merupakan interpretasi kosong untuk suatu kalimat F jika I tidak memberi nilai kebenaran pada proposisi apapun. Untuk membedakan interpretasi yang satu dengan interpretasi yang lain diberi notasi huruf I yang diikuti dengan bilangan asli.

Definisi 2.6 :

1. Kalimat F disebut absah bila F bernilai benar terhadap setiap interpretasi untuk F, F juga disebut *Tautologi*
2. Kalimat F disebut terpenuhi bila F bernilai benar terhadap suatu interpretasi untuk F.
3. Kalimat F disebut kontradiksi bila F bernilai salah terhadap setiap interpretasi untuk F.
4. Kalimat F dikatakan berakibat kalimat G bila untuk setiap interpretasi I, F benar terhadap I maka G benar terhadap I
5. Kalimat F dan G dikatakan setara bila untuk setiap interpretasi I maka F dan G bernilai sama terhadap I.

6. Kalimat F_1, F_2, \dots, F_n dikatakan konsisten bila terdapat suatu interpretasi I sedemikian sehingga F_1, F_2, \dots, F_n bernilai benar terhadap I

Teorema 2.1 :

F merupakan kalimat terpenuhi bila non F kalimat tidak absah

Teorema 2.2 :

F merupakan kalimat kontradiksi bila non F kalimat absah

Untuk menentukan keabsahan suatu kalimat dikenal cara-cara sebagai berikut :

1. Menggunakan tabel kebenaran

Dibuat tabel kebenaran dengan judul-judul kolom anak kalimat pada kalimat yang akan diselidiki, dibuat proposisi kalimat, selanjutnya tabel dibentuk dengan menampung semua kemungkinan interpretasi yang ada.

2. Menggunakan pohon semantik

Vertek-vertik pohon semantik diberi nomor mulai dari nomor 1.

Cabang di vertek 1(v_1) menyatakan nilai kebenaran dari salah satu proposisi. Cabang tersebut menghasilkan v_1 dan v_3 , kemudian diadakan analisa terhadap v_2 dan v_3 dan seterusnya sampai semua proposisi yang

vertik-vertiknya sesuai dengan cabang-cabang yang ada selesai dianalisa.

Contoh 2.3 :

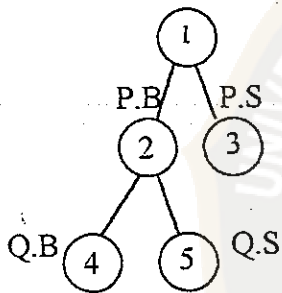
$$E : \underline{(\neg P \vee \neg Q)} \Rightarrow (\neg (\underline{P \wedge Q}))$$

F

G

H

Pohon semantiknya adalah :



Keterangan :

- P.B : P bernilai benar
- P.S : P bernilai salah
- Q.B : Q bernilai benar
- Q.S : Q bernilai salah

i. analisa vertek 2(av2) : tidak menghasilkan apa-apa

ii. av3 : P bernilai salah bila G salah

G salah bila H benar

bila E benar

iii. av4 : P benar, Q benar bila $(\neg P)$ salah, $(\neg Q)$ salah

bila F salah

bila E benar

iv. av5 : P benar, Q salah bila G salah

bila H benar

bila E benar

Jadi E absah

3. Menggunakan alur kontradiksi

Digunakan pengandaian ingkaran dari yang akan dibuktikan, selanjutnya menggunakan langkah-langkah yang benar sehingga diperoleh suatu keadaan kontradiksi.

Contoh 2.4 :

Buktikan bahwa $E : (\neg P \vee \neg Q) \Rightarrow (\neg (P \wedge Q))$ absah !

F	G
—————	
H	

Andaikan E tidak absah

 bila F benar, H salah

 H salah bila G benar

 bila P benar, Q benar

 bila $\neg P$ salah, $\neg Q$ salah

 bila F salah.

KONTRADIKSI

Jadi E absah

2.1.2 Predikat kalkulus

Misal diberikan kalimat logika proposisi sebagai berikut :

E : Untuk semua jenis monitor komputer dapat digunakan atau ada monitor komputer yang tidak bisa digunakan.

Terlihat bahwa bahasa kalimat logika proposisi tidak cukup untuk mengungkapkan semua obyek seperti yang terdapat pada kalimat di atas. Kalimat nyata di atas dapat dinyatakan :

E : (untuk semua x) (jika x monitor komputer maka x dapat digunakan)

atau

E : (untuk suatu x) (x monitor komputer dan x tidak bisa digunakan)

sehingga bentuk abstrak dari kalimat E adalah :

$P(x) = x$ monitor komputer

$Q(x) = x$ monitor yang bisa digunakan

Bahasa yang merupakan perluasan dari bahasa proposisi di atas disebut bahasa logika predikat. Di mana :

- i. (untuk semua x) disebut kuantor universal dengan notasi ' \forall '
- ii. (untuk suatu x) disebut kuantor eksistensial dengan notasi ' \exists '

2.1.2.1. Aturan Sintaks Bahasa Logika Predikat

1. Kalimat logika predikat dibentuk oleh simbol-simbol seperti di bawah ini :

- a. nilai kebenaran : benar dan salah
- b. konstanta : a, b, c, a', b',... dan sebagainya
- c. peubah : x, y, z, x', y',... dan sebagainya
- d. fungsi : f, g, h, ... dan sebagainya
- e. predikat : P, Q, R,... dan sebagainya

2. Bahasa logika predikat dibentuk melalui 3 (tiga) tahap, yaitu :

- a. pembentukan term

- b. pembentukan proposisi
- c. pembentukan kalimat

Definisi 2.7:

Term merupakan ekspresi yang menyatakan obyek

Contoh 2.5:

a merupakan term berbentuk konstanta, $f(a,x)$ term berbentuk fungsi yang memuat konstanta a dan peubah x disebut juga fungsi dengan *arity* 2.

Definisi 2.8 :

Proposisi dalam logika predikat menyatakan sebuah relasi antar obyek.

Definisi 2.9 :

Operator-operator penghubung seperti pada logika proposisi juga berlaku pada logika predikat.

Definisi 2.10 :

Ekspresi dalam logika predikat adalah term atau kalimat

Definisi 2.11:

Term yang dipakai membentuk suatu term atau kalimat dinamakan subterm, sedangkan kalimat yang membentuk suatu term atau kalimat dinamakan subkalimat. Subterm dan subkalimat juga disebut subekspresi.

2.1.2.2. Aturan Semantik Bahasa Logika Predikat

Aturan-aturan semantik untuk logika predikat meliputi :

1. Aturan pemberian nilai konstanta, peubah, fungsi maupun predikat sesuai pembahasan dengan memperhatikan domainnya.
2. Aturan benar dan salah juga berlaku
3. Aturan penghubung logika dilakukan antar term
4. (untuk semua x) F bernilai benar jika untuk setiap elemen $d \in D$, F akan bernilai benar terhadap perluasan interpretasi $(x \leftarrow d)_0 I$ atau berarti kalimat F bernilai salah jika terdapat elemen $d \in D$ sedemikian sehingga F bernilai salah terhadap perluasan interpretasi $(x \leftarrow d)_0 I$.
5. (untuk suatu x) F bernilai benar jika terdapat elemen $d \in D$ sedemikian sehingga F bernilai benar terhadap perluasan interpretasi $(x \leftarrow d)_0 I$, atau berarti kalimat F bernilai salah jika untuk setiap elemen $d \in D$ sedemikian sehingga F bernilai salah terhadap perluasan interpretasi $(x \leftarrow d)_0 I$.

Cara menentukan keabsahan kalimat logika predikat dapat dilakukan dengan langkah-langkah sebagai berikut :

1. Dalam kalimat logika predikat umumnya memuat kuantor, maka langkah pertama adalah membuat analisa benar terhadap anak kalimat ruas kiri dengan terlebih dulu memperhatikan kuantornya.

2. Analisa tersebut diarahkan ke bentuk anak kalimat ruas kanan sehingga nilai kebenaran anak kalimat ruas kanan dapat ditentukan.
3. Dengan memperhatikan aturan semantik penghubung logika yang ada akan dapat ditentukan absah tidaknya kalimat tersebut.

Contoh 2.6:

Tunjukkan E : $\neg (\forall x) P(x) \Leftrightarrow (\exists x) (\neg P(x))$ absah !

1. Andaikan $\neg (\forall x) P(x)$ bernilai benar, tepat bila $(\forall x) P(x)$ bernilai salah, tepat bila terdapat $d \in D$ sedemikian sehingga $P(x)$ bernilai salah terhadap perluasan interpretasi $(x \leftarrow d) \circ I$.
2. Karena terdapat $d \in D$ sedemikian sehingga $P(x)$ bernilai salah tepat bila $\neg P(x)$ bernilai benar. Terdapat $d \in D$ sedemikian sehingga $\neg P(x)$ bernilai benar terhadap perluasan interpretasi $(x \leftarrow d) \circ I$, tepat bila $(\exists x) (\neg P(x))$ bernilai benar terhadap perluasan interpretasi $(x \leftarrow d) \circ I$.
3. Sesuai aturan semantik operator penghubung biimplikasi (\Leftrightarrow), maka benar \Leftrightarrow benar, tepat bila E bernilai benar, tepat bila E absah.

2.2. HIMPUNAN

Himpunan adalah kumpulan obyek atau benda yang mempunyai kriteria berbeda dan dirinci dengan jelas. Elemen- elemen himpunan termuat dalam tanda kurung kurawal, $\{ \}$. Himpunan dapat dinyatakan dengan menyebutkan secara

lengkap anggota-anggotanya atau dengan menyebutkan sifat yang harus dimiliki oleh himpunan tersebut dengan bentuk umum sebagai berikut :

$$\{ \text{deklarasi} \mid \text{predikat} \}$$

deklarasi menunjukkan elemen dan predikat menunjukkan sifat yang harus dimiliki himpunan tersebut.

a. Elemen Himpunan

Untuk menyatakan suatu elemen dari suatu himpunan digunakan notasi \in .

Jika diberikan $a \in A$, diartikan a adalah anggota himpunan A . Sebaliknya untuk menyatakan *bukan anggota* himpunan digunakan notasi \notin .

b. Himpunan Kosong

Himpunan kosong adalah himpunan yang tidak memiliki anggota dan dinotasikan dengan $\{\}$ atau \emptyset .

c. Kesamaan Himpunan

Kesamaan himpunan dinotasikan dengan tanda " $=$ ". Jika terdapat dua himpunan A dan B , maka $A=B$ dikatakan bernilai benar jika himpunan A dan B mempunyai anggota-anggota yang sama.

d. Himpunan Bagian

Jika terdapat dua himpunan A dan B , dan $A \subset B$ dengan anggota himpunan A juga merupakan anggota dari himpunan B maka himpunan A disebut

himpunan bagian dari himpunan B dan dinotasikan $A \subseteq B$. Tetapi jika $A \neq B$, himpunan A disebut himpunan bagian sejati dari himpunan B dan dinotasikan dengan $A \subset B$.

Himpunan seluruh himpunan bagian dari sebuah himpunan A dinotasikan dengan $\mathcal{P}A$ yang menyatakan himpunan kuasa dari A.

e. Gabungan Himpunan dan Irisan Himpunan

Gabungan himpunan merupakan himpunan yang beranggotakan sekurang-kurangnya menjadi anggota salah satu himpunan tersebut. Gabungan himpunan dinotasikan dengan operator “ \cup ”.

Irisan himpunan merupakan himpunan yang beranggotakan elemen-elemen yang sekaligus berada dalam kedua himpunan tersebut. Irisan himpunan dinotasikan dengan “ \cap ”.

Contoh 2.7 :

Jika ada dua himpunan A : { 1, 2, 3, 4 } dan himpunan B : { 3, 4, 5 }, maka :

$$A \cup B : \{1, 2, 3, 4, 5\}$$

$$A \cap B : \{3, 4\}$$

f. Selisih Himpunan

Selisih dua himpunan A dan B adalah himpunan yang beranggotakan elemen-elemen dalam A yang tidak termasuk dalam B, dan dinotasikan

dengan “ $A-B$ ”.

g. Hasil Ganda Kartesius (*Cartesian-product*)

Cartesian-product dari dua himpunan A dan B dinotasikan dengan $A \times B$ yang merupakan pasangan berurutan dimana elemen pertama adalah anggota himpunan A dan elemen kedua anggota himpunan B.

Contoh 2.8 :

$$\{a, b\} \times \{1, 2, 3\} = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3)\}$$

h. Kardinalitas Himpunan

Kardinalitas dari suatu himpunan adalah banyaknya anggota dari himpunan tersebut. Kardinalitas dilambangkan dengan “# ...” mendahului penulisan himpunan.

Contoh 2.9 :

{file, komp, monitor, disket}, maka kardinalitas dari himpunan adalah 4.

2.3. RELASI DAN FUNGSI

2.3.1. Relasi

a. Relasi sebagai himpunan pasangan terurut

Relasi adalah himpunan pasangan terurut dan bisa merupakan hasil ganda kartesius dari dua himpunan. Jika diberikan dua himpunan A dan B, maka didefinisikan :

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$$

Apabila relasi tersebut adalah R maka, dapat ditulis aRb atau $R(a, b)$.

Contoh 2.10 :

Misalkan diberikan dua himpunan $A = \{a, b\}$ dan $B = \{x, y\}$, maka :

$$A \times B = \{(a, x), (a, y), (b, x), (b, y)\} \text{ dan}$$

b. Relasi invers

Setiap relasi R dari A ke B mempunyai suatu invers R^{-1} dari B ke A yang didefinisikan oleh :

$$R^{-1} = \{(b, a) \mid (a, b) \in R\}$$

Dengan kata lain, relasi invers R^{-1} terdiri atas pasangan-pasangan terurut yang elemen-elemennya dipertukarkan letaknya.

c. Domain dan range suatu relasi

Misalkan R suatu relasi dari A ke B, maka domain dari relasi R adalah himpunan semua elemen pertama dari himpunan pasangan terurut yang termasuk dalam relasi R, yaitu :

$$\text{dom} = \{a \mid a \in A, (a, b) \in R\}$$

Range atau jangkauan dari relasi R adalah himpunan semua elemen ke dua yang muncul dalam himpunan pasangan terurut, yaitu :

$$\text{rng} = \{b \mid b \in B, (a, b) \in R\}$$

2.3.2 Fungsi

Suatu relasi R dari A ke B di sebut sebagai fungsi jika setiap elemen A berhubungan hanya sekali dengan elemen dari B . Fungsi juga disebut sebagai pemetaan, sehingga R adalah pemetaan dari himpunan A ke dalam himpunan B .

Jenis fungsi dapat dibedakan menjadi :

a. Fungsi injeksi

Misalkan fungsi F memetakan A ke dalam B , maka F disebut fungsi injeksi jika setiap elemen A yang berbeda ditetapkan dengan setiap elemen B yang berbeda pula.

b. Fungsi surjeksi

Misalkan fungsi F memetakan A ke dalam B , maka F disebut fungsi surjeksi jika semua elemen dari B berpasangan dengan elemen dari A .

c. Fungsi bijeksi

Fungsi bijeksi merupakan gabungan dari kedua fungsi injeksi dan surjeksi.

d. Fungsi identitas

Misalkan A adalah sebarang himpunan, maka fungsi F yang memetakan A ke dalam A disebut fungsi identitas. Dalam perkataan lain fungsi F memetakan setiap elemen A pada elemen-elemen itu sendiri.

2.4. SPESIFIKASI PERANGKAT LUNAK

Dalam pengembangan perangkat lunak, faktor utama yang menentukan adalah adanya pernyataan-pernyataan persyaratan dari pengguna. Pernyataan persyaratan ini merupakan deskripsi informal tentang obyek-obyek pada sistem baru atau deskripsi tambahan pada sistem yang telah ada. Secara umum biasanya ditulis dengan menggunakan bahasa natural. Pernyataan persyaratan sangat memerlukan pemrosesan yang seksama sebelum menjadi dokumen acuan bagi perancang sistem.

2.4.1. Pernyataan Persyaratan

Pernyataan persyaratan dalam spesifikasi perangkat lunak terdiri dari :

a. Persyaratan fungsional

Merupakan pernyataan yang berisi persyaratan terperinci yang menyatakan tentang apa yang harus dilakukan oleh sistem.

b. Persyaratan non-fungsional

Merupakan pernyataan yang berisi batasan-batasan (*constraints*) untuk sistem bilamana sistem dioperasikan dan batasan-batasan bagi keleluasaan perancang dalam melakukan kegiatan pengembangan.

Dalam merumuskan garis besar persyaratan akan sulit untuk mengharapkan definisi persyaratan yang pasti sebelum kegiatan pengembangan sistem dimulai. Untuk itu diperlukan tahap-tahap tertentu untuk mendapatkan persyaratan yang diinginkan.

Di bawah ini adalah tahapan-tahapan yang dilalui:

a. Studi kelayakan (Feasibility Study)

Melakukan perkiraan apakah sistem yang diusulkan oleh pengguna dapat dipenuhi dengan teknologi perangkat lunak dan perangkat keras yang ada sekarang, jika ditinjau dari segi ekonomi apakah akan menghasilkan keuntungan dan jika diperlukan dapat mempelajari pengetahuan di luar lingkup sistem.

b. Menangkap persyaratan dan analisis

Proses mendapatkan persyaratan dengan cara melakukan pengamatan terhadap sistem yang telah ada.

c. Definisi persyaratan

Merupakan pernyataan berbahasa natural yang menyatakan apa yang diinginkan pengguna mengenai sistem. Dokumen ini dirumuskan untuk dijadikan basis bagi deskripsi abstrak sistem persyaratan.

d. Spesifikasi persyaratan

Deskripsi secara rinci dan tepat untuk dasar penyusunan desain dan implementasi. Deskripsi dapat mengalami perubahan sebagai konsekuensi dari kesalahan-kesalahan yang mungkin muncul.

Kegiatan utama dalam studi kelayakan adalah ketika harus memilah-milah persyaratan ke dalam persyaratan fungsional, persyaratan non-fungsional dan data persyaratan. Setiap persyaratan harus bisa ditentukan secara jelas dan tidak rancu.

Kemudian tujuan-tujuan yang diprioritaskan mulai diidentifikasi dan ditulis secara tepat dan lengkap. Tujuan pengembangan sistem harus ditentukan atau dibuat peringkat sehingga dapat diketahui sasaran yang ingin dicapai terlebih dulu.

