Nama: Yuliza Rahmi

Kelas : 1 TIPS 4

**BAB 7 : SISTEM BASIS BILANGAN**

* 1. **TEORI BILANGAN**

Terdapat 4 cara utama dalam penyajian balangan.

* + 1. **Bilangan Desimal**

Bilangan desimal adalah sistem bilangan yang berbasis sepuluh. Bilangan tersebut terdiri dari 0 sampai dengan 9. Beriku ini beberapa contoh bilangan dalam bentuk desimal:

* + 1. **Bilangan Biner**

Bilangan dalam bentuk biner adalah bilangan berbasis 2 yang digunakan sebagai standar bahasa yang digunakan oleh komputer. Ini menyatakan bahwa bilangan yang terdapat dalam sistem ini hanya 0 dan 1. Berikut contoh dari bilangan biner:

* + 1. **Bilangan Oktal**

Bilangan oktal adalah bilangan yang berbasis 8. Bilangan ini tersiri dari 0-7. Berikut contoh penulisan bilangan oktal:

* + 1. **Bilangan Heksadesimal**

Merupakan sistem bilangan berbasis 16. Sistem ini hanya memperbolehkan penggunaan bilangan dalam skala 0-9, dan menggunakan huruf A-F, atau a-f karena perbedaan kapital huruf tidak memiliki efek apapun.

* + 1. **Kode yang Mewakili Data**

Suatu komputer yang berbeda menggunakan kode binari yang berbeda untuk mawakili suatu karakter. Komputer yang 1 byte terdiri dari 4 bit, menggunakan kode binari yang berbentuk kombinasi 4 bit. Yaitu BCD(Binary Coded Decimal). Komputer yang 1 byte terdiri dri 6 bit, menggunakan kode binari yang berbentuk 6 bit, yaitu SBCDIC(Standard Binary Coded Decimal Interchange Code). Komputer yang 1 byter terdiri dari 8 bit, mengguakan kode binari yang berbentuk kombinasi 8 bit, yaitu EBCDIC(Extende Binary Coded Decimal Interchange Code) atau ASCII(American Standart Code for Information Interchange).

1. **BCD(Binary Coded Decimal)**

Merupakan kode bnari yang dipergunakan hanya untuk mewakili nilai desimal saja, yaitu nilai 0-9. BCD menggunakan kombinasi dari 4 bit, sehingga sebanyak 16(24=16) kemungkinan kombinasi yang bisa diperoleh dan hanya 10 kombinasi yang dipergunakan.

0000 = 0

0001 = 1

0010 = 2

0011 = 3

0100 = 4

0101 = 5

0110 = 6

0111 = 7

1000 = 8

1001 = 9

1010 = A (tidak boleh)

1011 = B (tidak boleh)

1100 = C (tidak boleh)

1101 = D (tidak boleh)

1110 = E (tidak boleh)

1111 = F (tidak boleh)

Kode BCD dipergunakan pada komuter generasi pertama dan kini sudah jurangdipergunakan untuk komputer generasi sekarang.

1. **SBCDIC(Standard Binary Coded Decimal Interchage Code)**

Menggunakan kombinasi 6-bit, sehingga lebih banyak kombinasi yang bisa dihasilkan, sebanyak 64(26=64) kombinasi kode, yaitu 10 kode untuk digit angka, 26 kode untuk huruf alphanumerik dan sisanya karakter-karakter khusus yang dipilih.

1. **EBCDIC(Extended Binary Coded Decimal Interchange Coded)**

Terdiri dari kombinasi 8 bit yang memungkinkan untuk mewakili karakter 256(28 = 256) kombinasi karakter. Pada EBCDIC, high-order atau 4 bit pertama desebut dengan zone bits dan low order bits atau 4 bit kedua disebit dengan numeric bits.

1. **Kode ASCII(American Standard Coe for Information Interchange)**

Kode ini bernilai dari 0-127(128 kode) meskipun tidak semuanya dapat ditampilkan di dalam program. Kode bernilai sampai dengan 31 dinamakan kode control, sedangkan 32-126 adalah kode ASCI yang dapat ditampilkan. Alfabet a-Z diwakili oleh kode desimal 65-90.

1. ASCII(American Standard Code for Information Interchange) 7-bit

ASCII dikembangkan oleh ANSI(american National Stanards Institute) untuk tujuan membuat kode biner yang standar. Kode ASCII yang standar menggunakan kombinasi 7-bit, dengan kombinasi sebanyak 127 dari 128(27=128) kemungkinan kombinasi yaitu:

* 26 buah huruf kapital(upper case) dari A-Z
* 26 buah huruf kecil(lower case) dari a-z
* Digit desimal dari 0-9
* 34 karakter kontrol yang tidak dapat dicetak hanya digunakan untuk informasi status operasi komputer
* 32 karakter khusus(special characters)
* ASCII 7-bit banyak digunakan untuk komputer-komputer generasi sekarang, termasuk komputer mikro

1. ASCII 8-bit

ASCII 8-bit terdiri dari kombnasi 8-bit mulai banyak digunakan, karena lebih banyak memberikan kombinasi karakter. Dengan ASCII 8-bit, karakter-karakter gfaphic yangtidak dapat iwakili ASCII 7-bit, seperti ♠ ♣ ♥ ♦ β α ► ◄ karakter dan sebagainya dapat diwakili. Komputer IBM PC menggunakan ASCII 8-bit.

* 1. **KONVERSI**
     1. **Konversi Desimal ke Biner / Biner ke Desimal**

Untuk mengubah angka desimal menjai angka biner digunakan metode pembagian dengan angka 2 sambil memperhatikan sisanya. Ambil hasil bagi dari proses pembagian sebelumnya, dan bagi kembali bilangan tersebut dengan 2. Ulangi lagkah-langkah tersebut hingga hasil bagi akhir bernilai 0 atau 1. Kemudian susun nilai-nilai sisa imulai dari nilai sisa terakhir sehingga diperoleh bentuk biner dari angka bilangan tersebut. Sebegai contoh:

1. **19210 = ................. 2 ?**

**Proses:**

192 : 2 = **96 sisa 0**

96 : 2 = **48 sisa 0**

48 : 2 = **24 sisa 0**

24 : 2 = **12 sisa 0**

12 : 2 = **6 sisa 0**

6 : 2 = **3 sisa 0**

3 : 2 = **1 sisa 1**

Hasilnya adalah nilai sisa yang diurutnkan dari terahir samapai awal proses, yaitu 110000002

1. **12610 = ................ 2 ?**

**Proses:**

126 : 2 = **63 sisa 0**

63 : 2 = **31 sisa 1**

31 : 2 = **15 sisa 1**

15 : 2 = **7 sisa 1**

7 : 2 = **3 sisa 1**

3 : 2 = **1 sisa 1**

Hasilnya adalah nilai sisa yang diurutkan dari terakhir samapai awal proses, yaitu 1111102

Konversi bilangan biner ke desimal disapatkan dengan menjumlahkan perkalian semua bit biner dengan perpangkatan 2 sesuai dengan bit tersebut. Sebagai contoh:

1. **110000002 = ................ 10 ?**

**Proses:**

11000000 = ( 1 x 27 ) + ( 1 x 26 ) + ( 0 x 25 ) + ( 0 x 26 ) + ( 0 x 25 ) + ( 0 x 24 ) + ( 0 x

23 ) + ( 0 x 22 ) + ( 0 x 21 ) + ( 0 x 20 )

= 128 + 64 + 0 + 0 + 0 + 0 + 0 + 0

= 19210

1. **1100112** = **.................... 10 ?**

**Proses:**

110011 = ( 1 x 25 ) + ( 1 x 24 ) + ( 0 x 23 ) + ( 0 x 22 ) **+** ( 1 x 21 ) + ( 1 x 20 )

= 32 + 16 + 0 + 0 + 2 + 1

= 5110

* + 1. **Konversi desimal ke Oktal / Heksadesimal dan Oktal / Heksadesimal ke Desimal**

Pengubahan bilangan desimal ke bilangan oktal atau bilangan heksadesimal pada dasarnya sama dengan konversi bilangan desimal ke biner. Perbedaannya terletak pada bilangan pembagi. Jika pada konversi biner pembaginya adalah angka 2, maka pada konversi oktal pembaginya adalah angka 8, sedangkan pada konversi heksadesimal pembaginya adalah 16.

Sebagai contoh konversi Desimal ke Okta**l**

1. **12610 = ......... 8 ?**

**Proses:**

126 : 8 =  **15 sisa 6**

5 : 8 =  **1 sisa 7**

Hasilnya adalah nilai sisa yang diurutkan dari terakhir sampai awal proses, yaitu 1768.

1. **19210 = ........... 8 ?**

**Proses:**

192 : 8 = **24 sisa 0**

24 : 8 = **3 sisa 0**

Hasilnya adalah nilai sisa yang diurutkan dari terakhir sampai awal proses, yaitu 3008.

Sebagai contoh konversi desimal ke heksadesimal:

1. **19210 = ................. 16 ?**

**Proses:**

192: 16 = **12 sisa 0**

Hasilnya adalah nilai sisa yang diurutkan dari terakhir sampai awal proses, yaitu C016.

1. **12610 = ................... 16 ?**

**Proses:**

126 : 16 = 7  **sisa 14**

Hasilnya adalah nilai sisa yan diurutkan dari terakhir sampai awal proses, yaitu 7E16.

Konversi bilangan Oktal dan Heksadesimal sama dengan konversi bilangan biner ke Desimal. Perbeaannya hanya terdapat pada penggunaan angka basis. Jika sestem biner menggunakan basis 2, maka pada bilangan Oktal, basis yang digunakan adalah 8 dan bilangan Heksadesimal adalah angka 16.

Sebagai contoh konversi Oktal ke Desimal:

**1768 = ........ 10?**

**Proses:**

176 = ( 1 x 82 ) + ( 7 x 81 ) + ( 6 x 80 )

= 64 + 56 + 6

= 126

Sebagai contoh konversi Heksadesimal ke Desimal:

1. **C016 = ....... 10 ?**

**Proses:**

C0 = ( C x 161 ) + ( 0 x 160 )

= 192 + 0

= 192

1. **7E16 = ......... 10 ?**

**Proses:**

7E = ( 7 x 161 ) + ( E x 160 )

= 112 +14

= 126

* + 1. **Konversi Biner ke Oktal dan Oktal ke Biner**

Untuk mengubah bilangan biner ke oktal, kita pilih bilangan tersebut menjadi 3 bit bilangan biner dari kanan ke kiri. Sebagai contoh:

**1011011112 = ...............8 ?**

**Proses:**

[( 1x22 )+( 0x21 )+( 1x20 )] [( 1x20 )+( 0x21 )+( 1x20 )] [( 1x22 )+( 1x21 )+( 1x20 )]

= [4+0+1] [4+0+1] [4+2+1]

= 55710

Mengubah sistem bialangan oktal menjadi bilangan biner dilakukan dengan cara kebalikan dari konversi biner ke oktal. Dalam hal ini masing-asing digit bilangan oktal diubah langsung menjadi bilangan biner dalam kelompok tiga bit, kemudian merangkai kelompok bit tersebut sesuai aturan semula. Sebagai contoh:

1. 5578  = ..........2 ?

Proses:

= 5 → 101, 5 → 101, 7 → 111

= 1011011112

1. 7078 = ......... 2 ?

Proses:

= 7 → 111, 0 → 000, 7 → 111

=1110001112

* + 1. **Konversi Biner ke Heksadesimal dan Heksadesimal ke Biner**

Pengubahan bilangan Biner ke Heksadesimal dilakukan dengan pengelompokan setiap empat bit Biner dimulai dari bit paling kanan. Kemudian konversikan setiap kelompok menjadi satu digit Heksadesimal. Sebagagai contoh:

011011112 = ............ 16 ?

Proses:

[(0 x 23)+(1x22)+(1x21)+(0x20)] [(1x23)+(1x21)+(1x20)]

= [0+4+2+0] [8+4+2+1]

= 6F16

Konversi bilangan Heksadesimal ke Biner dilakukan dengan membalik urutan dari proses pengubahan Biner ke Heksadesimal. Satu digit Heksadesimal dikonversi menjasi 4 bit Biner. Sebagai contoh:

6F16 = .......... 2 ?

Proses:

= 6 → 0110

= F → 1111

= 011011112

* + 1. **Operasi Aritmatika** 
       1. **Sistem Penjumlahan Bilangan Biner**

Prosedur penjumlahan bilangan biner sama dengan penjumlahan bilangan desimal, tetapi hasil penjumlahannya hanya dinyatakan dalam angka 0 dan 1. Proses langkah-langkahnya sebagai berikut:

1. Digit-digit dari bilangan-bialngan biner ditambahkan satu persatu mulai dari posisi kolom palng kanan.
2. Jika jumlah bilangan biner telah melebihi 1, maka dikurangi dengan 2 untuk dibawah(carry of) 1 ke pertambahan kolom berikutnya.
3. Empat kombinasi dalam penjumlahan bilangan biner:

0+0 = 0 carry 0 || 0+1 = 1 carry 0 || 1+0 = 1 carry 0 || 1+1 = 0 carry 1

Contoh:

20 = 10100

20 = 10100

---------------------------- +

40 101000

* + - 1. **Sistem Pengurangan Bilangan Biner**

Prosedur yang digunakan dalam penurangan bilangan biner juga sama dengan pengurangan bilangan desimal, hanya apabila yang dikurangi lebih kecil daripada pengurangnya, maka diperlukan borrow(pinjaman) dari kolom di sebelah kirinya. Empat kombinasi dalam pengurangan bilangan biner:

1. = 0 borrow 0 || 0-1 = 1 borrow 1 || 1-0 = 1 borrow 0 || 1-1 = 0 borrow 0

Sebagai contoh:

27 = 11011

9 = 1001

----------------------- -

18 10010

* + - 1. **Sistem Perkalian Bilangan Biner**

Perkalian bilangan binary dapat dilakukan dengan cara yang sama dengan perkalian bilangan desimal. Dasar perkalian untuk masing-masing digit bilangan binary adalah:

0x0 = 0 || 1x0 = 0 || 0x1 = 0 || 1x1 = 1

Hal-hal yang perlu do]iperhatikan dalam perkalian bilangan binary:

1. Jika pengali adalah bernilai 1, maka hasilnya berupa bilangan binary yang dikali, sehingga cukup disalin saja.
2. Jika pengali adalah bernilai 0 maka hasilnya semua adalah 0.

Sebagai contoh:

14 = 1110

12 = 1100

------------------------------- x

28 000 0 hasil dari 1110x0

1 4 000 0 hasil dari 1110x0

1100 hasil dari 1110x1

1110 hasil dari 1110x1

------------------------------- +

1 68 101010000

* + - 1. **Sistem Pembagian Bilangan Biner**

Pembagian bilangan binary juga dapat dilakukan dengan cara yang sama dengan pembagian desimal. Pembagian dengan digit binary 0 tidak mempunyai arti, sehingga dasar pembagian bilangan binary adalah:

0:1 = 0 || 1:1 = 1

Sebagai contoh:

5 / 125 \ 25

10

--------

25

* + - 1. **Sistem Penjumlahan Bilangan Oktal**

Prosedur penjumlahan bilangan oktal sama dengan penjumlahan desimal atau bilangan biner. Jika jumlah bilnangan oktal telah melebihi 7, kurangi jumlah tersebut dengan 8, lebih/sisanya ditulis dan carry 1 harus ditambahkan pada kolom di sebelah kirinya.

Contoh:

4448 + 2228 = .......... ?

Proses:

Kolom 1: 4+2 = 6

Kolom 2: 4+2 = 6

Kolom 3: 4+2 = 6

Hasil 4448 + 2228 = 6668

* + - 1. **Sistem Pengurangan Bilangan Oktal**

pengurangan bilangan oktal dapat dilakukan secara sama edngan penguranagn desimal. Hanya apabila yang dikurangi lebih kecil daripada pengurangannya, maka diperlukan borrow(pinjaman) dari kolom di sebelah kirinya.

Sebagai contoh:

108 = 1548

87 = 1278

--------------------- -

21

Proses:

Pinjam sebelah jadi 8+4-7=58

Karena dipinjam maka 5-2-1=28

Hasilnya adalah 258

* + - 1. **Sistem Perkalian Bilanangan Oktal**

Langkah-langkah dalam melakukan perkalian bialngan oktal:

1. Kalikan masing-masing kolom secara desimal
2. Ubah dari hasil desimal ke oktal
3. Tuliskan hasil dari digit paling kanan dari hasil oktal
4. Jika hasil perkalian tiap-tiap kolom terdiri dari 2 dijit, maka digit paling kiri merupakan carry of untuk ditambahkan pada hasil perkalian kolom selanjutnya.
   * + 1. **Sistem Pembagian Bilangan Oktal**

Contoh pembagian bilangan oktal:

12 / 168 \ 14 14 / 250 \ 16

12 14 → 148 x 18 = 148

----------- - -----------

48 110

48 110 → 148 x 68 = 48 x 68 = 308

---------- - -----------

0 0 = 18 x 68 = 68

1108

* + - 1. **Sistem Penjumlahan Bilangan Heksadesimal**

Jika jumlah bilagan heksadesimal telah melebihi 7, kurangi jumlah bilangan tersebut dengan 16, lebih/sisanya ditulis dan carry 1 harus ditambahkan pada kolo disebelah kirinya.

Contoh:

BAD16  + 43116 = .......... ?

Proses:

Kolom 1: D + 1 = E, artinya D simbol 13 dan E simbol 14

Kolom 2: A + 3 = D, artinya A simbol 10 dan D simbol 13

Kolom 3: B + 4 = f, artinya B simbol 11 dan F simbol 15

Hasil BAD16 + 43116 = FDE16

* + - 1. **Sistem Pengurangan Bilangan Heksadesimal**

Pengurangan bilangan heksadesimal dapat dilakukan secara sama dengan pengurangan desimal, hanya apabila yang dikurangi lebih kecil daripada pengurangnya, maka diperlukan borrow(pinjaman) dari kolom di sebelah kirinya.

Contoh:

4833 = 12E1

1575 = 627

-------- ---------

3258 CBA

* + - 1. **Sistem Perkalian Bbilangan Heksadesimal**

Perkalian bilangan heksadesimal dapat dilakukan secara sama dengan perkalian bilangan desimal dengan langkah-langkah sebagai berikut:

1. Mengalikan masing-masing kolom secara desimal
2. Ubah dari hasil desimal ke heksadesimal
3. Tuliskan hasil dari digit paling kanan dari hasil heksadesimal
4. Jika hasil perkalian tiap-tiap kolom terdiri dari 2 digit, maka digit paling kiri merupakan carry of untuk ditambahkan pada hasil perkalian kolom selanjutnya.

Contoh:

172 = AC16

27 = 1B16

------ --------- x

168 764

AC

-------

1224

* + - 1. **Sistem Pembagian Bilangan Heksadesimal**

Pembagian Heksadesimal dapat dilakukan dengan cara seperti bilangan desimal.

* + - 1. **Sistem Penjumlahan Bilangan BCD(Binary Code Decimal)**

Bilangan BCD adlah bilangan biner yang menyimbolkan bilangan desimal. Prosedur penjumlahan bulangan BCD:

1. Jumlahkan bilangan BCD seperti penjumlahan bilangan biner biasa.
2. Juka jumlahnya ≤ 9, maka jumlah tersebut adlah jawaban yang benar.
3. Juka jumlahnya > 9, lakukan langkah ke 4.
4. Tambahkan 6 (0110) pada hasil penjumlahan tersebut. Jumlahkan carry pada bilangan di sebelah kirinya.
5. Ulangi langkah 1 sampai 4 untuk setipa kelompok bit BCD.

Contoh:

1001 + 0101 = 9+5 = .....

1001BCD → 9

0101BCD → %

Hasil 1110BCD

Tambah 6 0110

-------- +

1 0100 = 0001 0100BCD = 1410

* + - 1. **Sistem Penjumlahan Bilangan BCD(Binary Code Decimal)**

Jika jumlah bilangan heksadesimal telah melebihi 15(F), kurangi jumlah tersebut dengan 16, lebih/sisanya ditlis dan carry 1 harus ditambahkan pada kolom di sebelah kirinya.

Contoh:

192 – 3 = .....

19210 1100 00002

310 0000 00112

--------- - --------------- -

18910 1011 11012 = 18910

* + - 1. **Metode Komplemen**

Metode komplemen 1 merupakan metode yang sederhana, proswes dilakukan dengan membalik(invers) tipa-tiap bit. Misal dalam sistem bilangan 8 bit, bilangan positif dimulai dari 0000 0000 – 0111 1111 = 0 – 127 bilangan negatif dimulai dari 1111 1110 – 1000 0000 = -1 – (-128). Langkah-langkah untuk mengkonversi dari bilangan desimal ke bilangan komplemen -1:

1. Jika bilangan desimal positif, bilangan komplemen -1 adalah bilangan biner biasa.
2. Jika nilangan desimal negatif, bilangan komplemen -1 dicari dengan cara
3. Mengkomplemenkan setiap bit dalam bilangan biner.

Contoh:

-3510 = .....

Bilangan biner: 0010 0011

Komplemen -1: 1101 1100

* + - 1. **Metode Komplemen 2**

Metode komplemen 2 merupakan metode yang paling banyak digunakan untuk operasi aritmatika bilangan biner dalam sistem komputer. Sistem komputer ada yang menggunakan sistem bilangan 8-bit, yang berarti 28=256 bilangan, dan 16 bit, yang berarti ada 216 = 65536 bilangan.

Untuk melambangkan bilangan positif dan negatif, metode komplemen -2 menggunakan MSB sebagai tanda (sign bit).

Langkah-langkah untuk mengkonversi dari bilangan desimal ke bilangan komplemen -2:

1. Jika bilangan desimal positif, bilangan komplemen -2 adalah bilangan biner biasa.
2. Jika bilangan desimal negatif, bilangan komplemen -2 dicari dengan cara:
3. Mengkomplemenkan setiap bit dalam bilangan biner untuk menjadi bilangan komplemen -1.
4. Menambahkan 1 pada bilangan komplemen -1 untuk memperoleh bit tanda.
   * + 1. **Sistem Perkalian**

Perkalian bilangan biner sama dengan perkalian bilagan desimal, teapi bilangan yang digunakan hanya 0 dan 1.

Contoh:

13 x 11 = .....

13 0000 1101

11 0000 1011

---- x ------------- x

13 0000 1101

13 00001 101

----- + 000000 00

143 0000110 1

----------------- +

0001000 11112 = 14310

* + - 1. **Sistem Pembagian**

Prosedur pembagian sama dengan prosedur perkalian.

**7.2.2 Logika**

**Pernyataan**

Untuk memahami pengertian tentang pernyataan simaklah contoh berikut ini:

Contoh:

1. Gaun itu indah
2. Nisa gadis yang lucu
3. Bronis kukus itu enak

Kalimat-kalimat diatas dapat bernilai benar saja atau bernilai salah saja, tetapi bersifat relative atau tergantung pada keadaan. Jadi, kalimat-kalimat seperti itu tidak dapat disebut sebagai pernyataa.

**7.2.2.1 Lambang dan Nilai Kebenaran Suatu Pernyataan**

Dalam matematika, pernyataan dengan huruf kecil seperti a, b, p dan q. Perhatika contoh berikut:

Contoh:

1. Pernyataan “7 adlah bilangan prima” dapat dilambangkan dengan huruf p, jadi p: 7 adalah bilangan prima.
2. Pernyataan “Ibu kota Jawa Timur adalah Surabaya” dapat dilambangkan dengan huruf q, jadi q: Ibu kota Jawa Timur adalah Surabaya.

**7.2.2.2 Mendeskripsikan, Inkaran, Konjungsi, Disjungsi, Implikasi, Biimplikasi dan**

**Ingkarannya.**

**Pernyataan Majemuk**

Apabila suatu pernyataan terdiri lebih dari 1 pernyataan maka diantara satu pernyataan lainnya dibutuhkan suatu kata penghubung suatu kata penghubung sehingga diperoleh suatu pernyataan majemuk

**Operasi Konjungsi**

Operasi merupakan operasi biner yang dilambangkan dengan tanda “ᴧ”. Dengan operasi ini dua pernyataan dihubungkan dengan kata “dan”.

Contoh: P = Guru hadir

Q = Murid tidak bersuka ria

P ᴧ Q = Guru hadir dan muri tidak senang

**Operasi Implikasi**

Operasi implikasi aalah operasi penggabungan dua pernyataan yang menggunakan kata hubung “jika .... maka ....” yang dilambangkan “=>”. Implikasi dari pernyataan p dan q ditulis p=>q adn ibaca “juka p maka q”. Pernyataan bernyarat p=>q juga dapat dibaca “p hanya jika q” atau “p adalah syarat cukup bagi q atau q adalah syarat bagi p”.

Dalam pernyataan p=>q

P disebut hipotesa/antesedan/sebab

Q disebut konklusi/konequen/akibat

**7.2.2.3 Mendeskripsikan Invers, Konvers dan Kontraposisi**

Dari suatu pernyataan bersyarat “p => q” yang diketahui dapat dibuat banyak pernyataan lain sebagau berikut:

1. q => p disebut pernyataan Konvers dari p => q
2. ~p => ~q disebut pernyataan Invers dari p => q
3. ~q => ~p disebut pernyataan Kontraposisi dari p => q

q => p (konvers) : jika saya tidak belajar maka lampu mati

~p => ~q (invers) : jika lampu tidak mati maka saya belajar

~Q => ~p (kontraposisi) : jika saya belajar maka lampu tidak mati

**7.2.2.4 Menerapkan Modus Ponens, Tollens dan Prinsip Silogisme Dalam Menarik**

**Kesimpulan**

Dalam bentuk implikasi, modus tollens dapat dituliskan sebagai [(p=>q) ᴧ~q]=>~p sah atau tidaknya mous tollens dapat diuji dengan tabel kebenaran sbb:

Tabel nilai kebenarans[(p=>q) ᴧ~q]=>~p

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p | q | ~p | ~q | p => q | (p => q) ᴧ ~ q | [(p => q) ᴧ ~ q] => ~p |
| B | B | S | S | B | S | B |
| B | S | S | B | S | S | B |
| S | B | B | S | B | S | B |
| S | S | B | B | B | B | B |

Dari tabel pada kolom 7 tampak bahwa [(p=>q) ᴧ~q]=>~p merupakan tautalogi. Jadi modus tollens merupakan argumentasi yang sah.

**Silogisma**

Dari premis-premis p => dan q => r dapat ditarik konklusi p => r. Penarikan kesimpulan seperti ini disebut silogisma. Skema argumnya dapat inyatakan sbb:

p => q ....... premis 1

q => r .......premis 2

p => r ....... kesimpulan/konklusi

Dalam bentuk implikasi, silogisme dapat dituliskan sebagai [(p=>q)ᴧ(q=>r)]=>(p=>r) sah atau tidaknya silogisme dapat diuji dengan tabel kebenaran sbb:

Tabel kebenaran [(p=>q) ᴧ(q=>r)]=>(p=>r)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| p | q | ~p | ~q | p => q | (p => q) ᴧ ~ p | [(p=>q) ᴧ(q=>r)]=>(p=>r) |
| B | B | S | S | B | S | B |
| B | S | S | B | S | S | B |
| S | B | B | S | B | B | S |
| S | S | B | B | B | B | B |

Dari tabel, pada kolom 7 tampak bahwa [(p=>q) ᴧ(q=>r)]=>(p=>r) bukan merupakan tautologi. Jadi argumentasi diatas tidak sah.