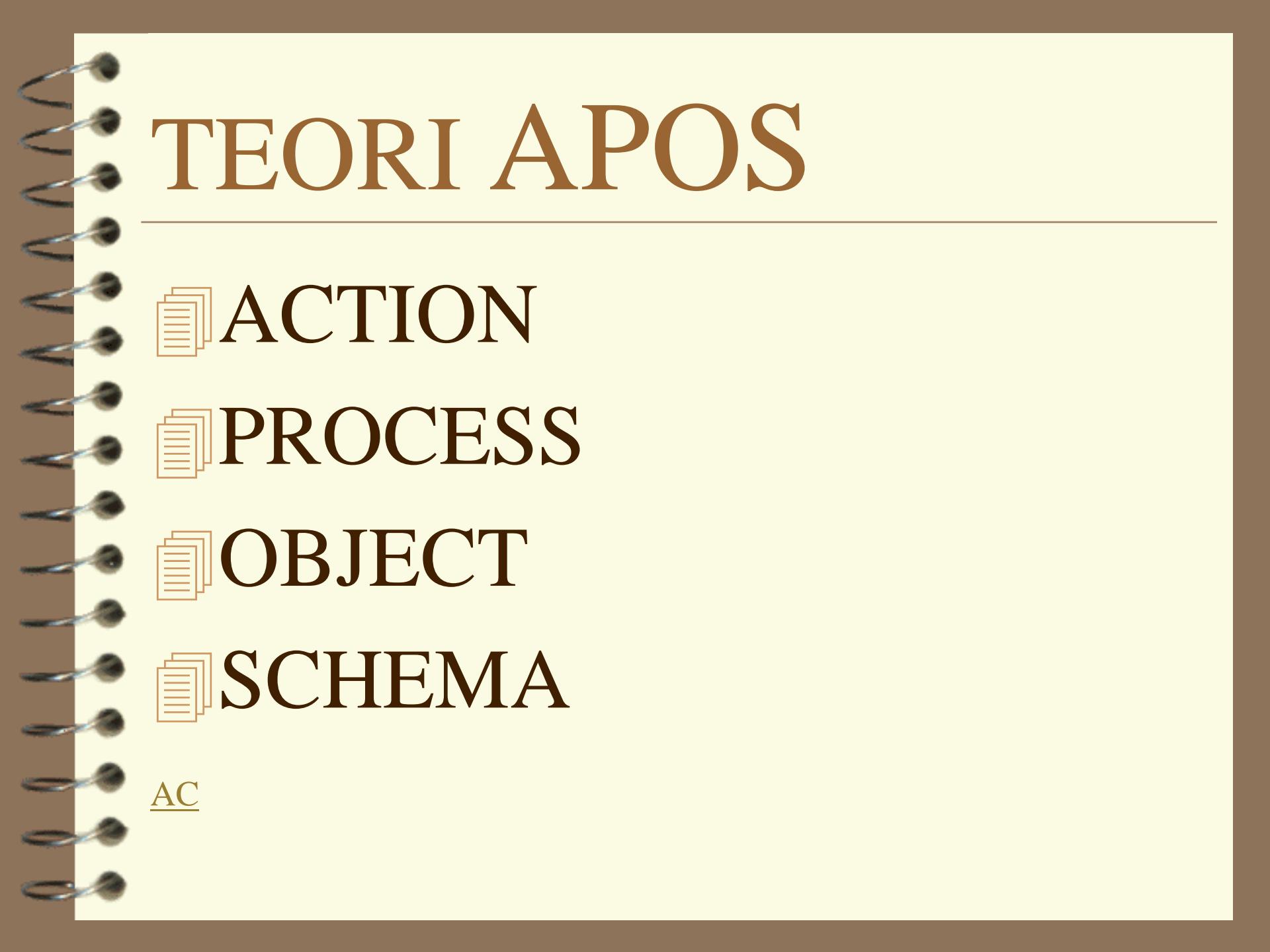


PENGGUNAAN PROGRAM ISETL DALAM PEMBELAJARAN ALJABAR

Oleh
Ealah Nurlaelah (UPI)
Ema Carnia (UNPAD)



TEORI APOS

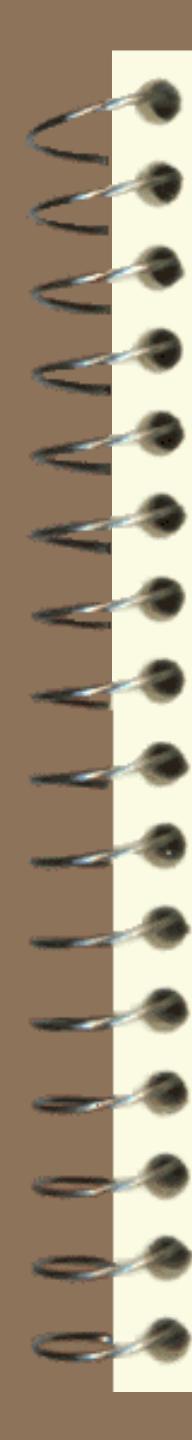
 ACTION

 PROCESS

 OBJECT

 SCHEMA

AC



SIKLUS ACE



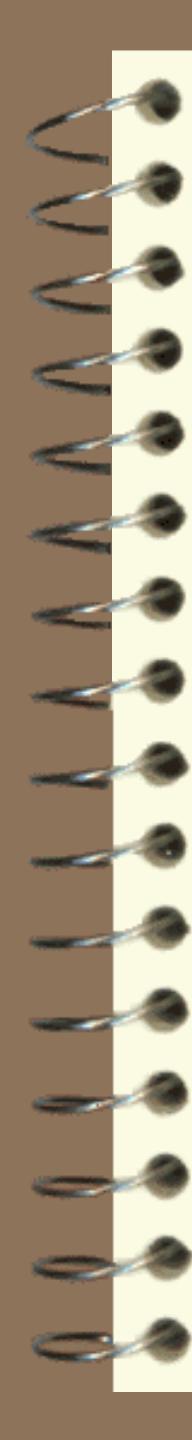
ACTIVITIES



CLASS DISCUSSION

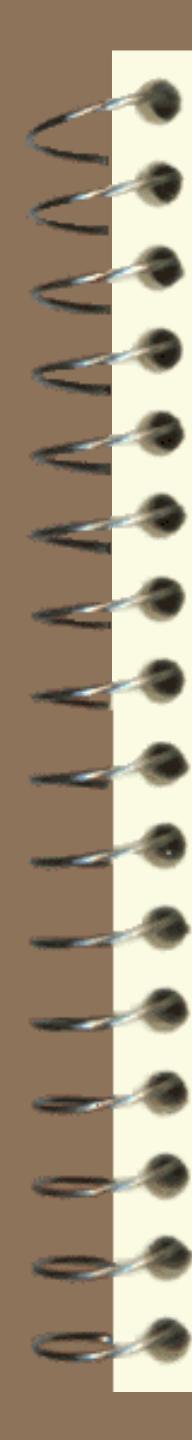


EXERCISES



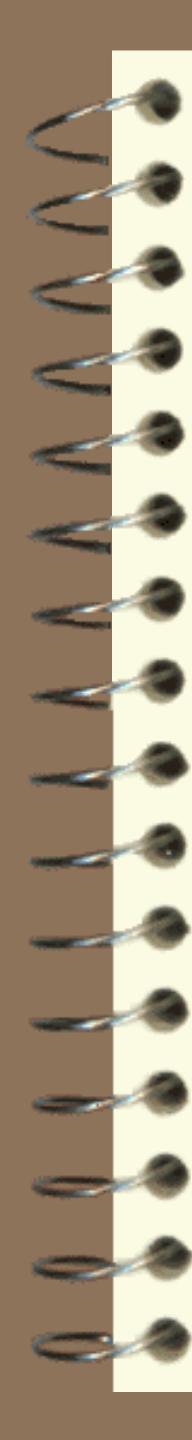
ACTION

 Suatu Tranformasi merupakan suatu *action* jika merupakan stimulus yang berasal dari luar



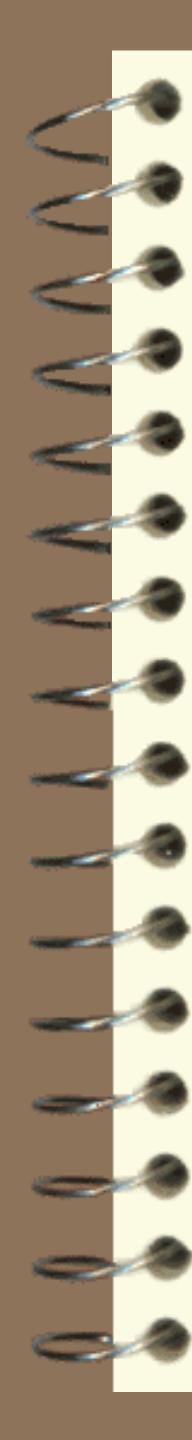
PROCESS

 Ketika seseorang merefleksikannya dan mengintegrasikannya maka action dapat menjadi bagian dari dirinya (internal) yang dapat dikontrol



OBJECT

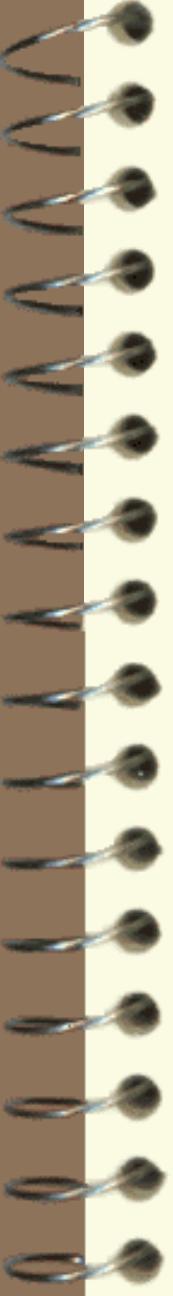
¶ Ketika individu menyadari suatu *process* sebagai suatu totalitas, menyadari bahwa transformasi dapat dilakukan padanya dan juga dapat mengkonstruksi transformasi tersebut, maka *process* sudah menjadi *object*



SCHEMA

 Koleksi dari *process* dan *object* dapat diorganisasikan dalam suatu struktur untuk membentuk suatu *schema*. Beberapa *Schema* dapat diperlakukan sebagai suatu *object* didalam *schema* yang lebih tinggi tingkatannya

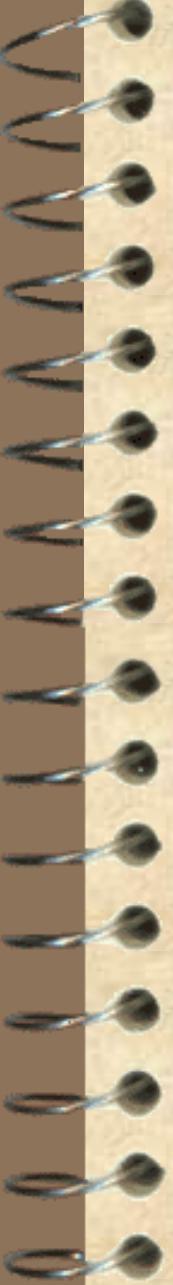
ACE



Beberapa Contoh

Pembelajaran dengan

Program ISETL.



Beberapa Instruksi ISETL sederhana

> 7+8;

25;

> 13 * (-233.8);

-3039.400;

> 6 = 2 * 3;

true;

> 5 >= 2 * 3;

false;

> 17

>> + 237 - 460

>> *2

>> ;

-513;

> n := 37 mod 23;

> n;

14;

> (2 /= 3) and ((5.2/3.1) > 0.9);

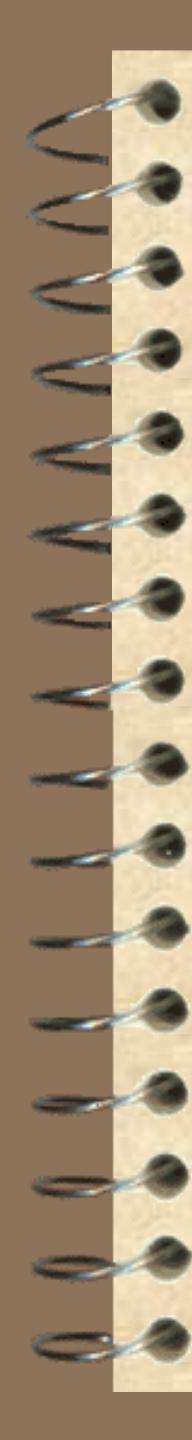
> (3 <= 3) impl (3 = 2 + 1);

Pemeriksaan Sifat-Sifat Ruang Vektor

Diketahui $Z_5 = \{0,1,2,3,4\}$ dengan
Operasi “ Penjumlahan Modulo 5 “
dan “ Perkalian Skalar Modulo 5 “
 $V := \{[a,b] | a,b \text{ in } Z_5\}$; Himpunan
pasangan terurut di Z_5

OPERASI PENJUMLAHAN SKALAR

```
>     ads := func(k,l);  
>> if k in Z5 and l in Z5 then  
>> return(k + l)mod 5;  
>> end;  
>> end;  
>     ads(4,4);  
3;
```



OPERASI PERKALIAN SKALAR

```
> ms := func(k,l);  
>> if k in Z5 and l in Z5 then  
>> return(k * l)mod 5;  
>> end;  
>> end;  
> ms(4,4);  
1;
```

OPERASI PERKALIAN SKALAR DENGAN VEKTOR

```
> sm := func(k,v);  
>>if k in Z5 then  
>>return[k * v(i) mod 5 : i in [1,2]];  
>>end;  
>>end;  
> sm(2,[4,4]);  
[3, 3];
```

OPERASI PENJUMLAHAN VEKTOR

```
> va := | v, w -> [(v(i) + w(i)) mod 5 : i in [1,2]]|;  
> va([3,4],[2,3]);  
[0, 2];
```

SIFAT TERTUTUP PADA PENJUMLAHAN VEKTOR

```
> is_closed_va := func(V,va);  
>>return forall v, w in V | v . va w in V;  
>>end;  
> is_closed_va(V,va);  
true;
```

SIFAT KOMUTATIF

```
> is_commun_va := func(V,va);  
>>return forall v, w in V | v .va w = w .va v;  
>>end;  
> is_commun_va(V,va);  
true;
```

SIFAT ASSOSIATIF

```
> is_assoc_va := func(V,va);  
  >> return forall u, v, w in V | (u .va v) .va w =  
    u .va (v .va w);  
  >> end;  
> is_assoc_va(V, va);  
true;
```

EKSISTENSI ELEMEN NOL

```
> has_zerovec := func(V,va);  
">>> VZERO := choose z in V | forall v in V |  
    ( v .va z) = v;  
>> return VZERO;  
>> end;  
> has_zerovec(V,va);  
[0, 0];
```

Memiliki Vektor Invers

```
> has_vinverses := func(V,va);  
  >> return  
  >> forall x in V | exists v in V | x .va v  
    = [0,0];  
  >> end;  
> has_vinverses(V,va);  
true;
```

SIFAT TERTUTUP PERKALIAN VEKTOR DENGAN SKALAR

```
> is_closed_sm := func(Z5, V, sm);  
>>return forall k in Z5, v in V| ( k .sm v)  in  
    V;  
>>end;  
> is_closed_sm(Z5, V, sm);  
true;
```

SIFAT ASSOSIATIF SKALAR TERHADAP VEKTOR

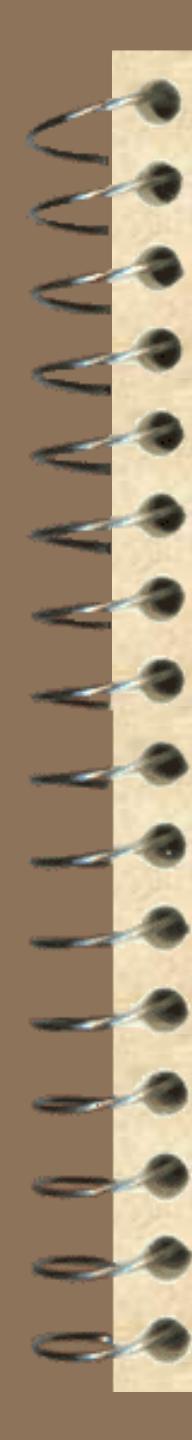
```
> is_assoc_sm := func(Z5, V, sm, ms);  
>>return forall s in Z5, t in Z5, v in V |  
    s .sm( t .sm v) = (s .ms t) .sm v;  
>>end;  
> is_assoc_sm(Z5, V, sm, ms);  
true;
```

SIFAT DISTRIBUTIF 1

```
> has_distributive1 := func(Z5, V, sm, va);  
  >> return forall s in Z5, v, w in V |  
        (s .sm(v .va w)) = (s .sm v).va (s.sm w);  
  >> end;  
> has_distributive1(Z5, V, sm, va);  
true;
```

SIFAT DISTRIBUTIF 2

```
> has_distributive2 := func(Z5, V, va, sm,  
    ads);  
>>return forall s,t in Z5, v in V |  
    (s .ads t) .sm v = (s .sm v) .va (t .sm v);  
>>end;  
> has_distributive2(Z5, V, va, sm, ads);  
true;
```



EKSISTENSI ELEMEN IDENTITAS

```
> has_identityscalar := func(Z5, V, sm);  
>>Identityscalar := choose e in Z5 |  
    forall v in V | ( e .sm v) = v;  
>>return Identityscalar;  
>>end;  
> has_identityscalar(Z5, V, sm);  
1;
```