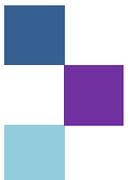


Pengantar Berpikir Komputasi dan Pemrograman Prosedural

Tim Pengajar KU1072
Sem. 1 2013-2014



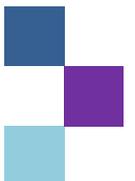
KU1072/Pengenalan Teknologi Informasi B
Tahap Tahun Pertama Bersama
Institut Teknologi Bandung





Tujuan Kuliah Pemrograman

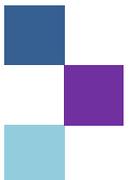
- Mahasiswa mampu:
 - Menjelaskan bagaimana komputasi/program dimanfaatkan dalam keilmuan fakultas/sekolah
 - Menjelaskan bagaimana proses dari source code menjadi program dengan menggunakan kompilator/interpreter
 - Menjelaskan apa yang dimaksud berpikir komputasi dengan pendekatan prosedural
 - Memahami representasi dan pemrosesan data dan program dalam mesin komputer
 - Membuat program kecil pertama dalam bahasa pemrograman yang dipilih dan memahami aspek eksekusinya”





Apa itu Berpikir Komputasi

- Berpikir menggunakan logika
 - Melakukan sesuatu selangkah demi selangkah
 - Menentukan keputusan bila menghadapi dua kemungkinan yang berbeda
- Salah satu cara untuk memahami dan mengimplementasikan cara berpikir komputasi adalah dengan belajar pemrograman
 - Cara lain?
 - Belajar memahami penggunaan perangkat lunak pengolah kata (contoh: MS Word), pengolah LembarKerja (SpreadSheet, contoh: MS Excel)





Kemampuan Berpikir Komputasi

“Berpikir komputasi memungkinkan kita untuk menggunakan komputasi sesuai dengan kebutuhan kita. Kemampuan ini akan menjadi salah satu kemampuan dasar yang harus dimiliki di abad 21”

(<http://link.cs.cmu.edu/article.php?a=600>)

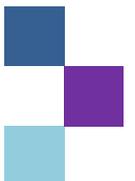
Kemampuan Dasar yang sudah ada sebelumnya:

- Kemampuan Membaca
- Kemampuan Menulis
- Kemampuan Berhitung

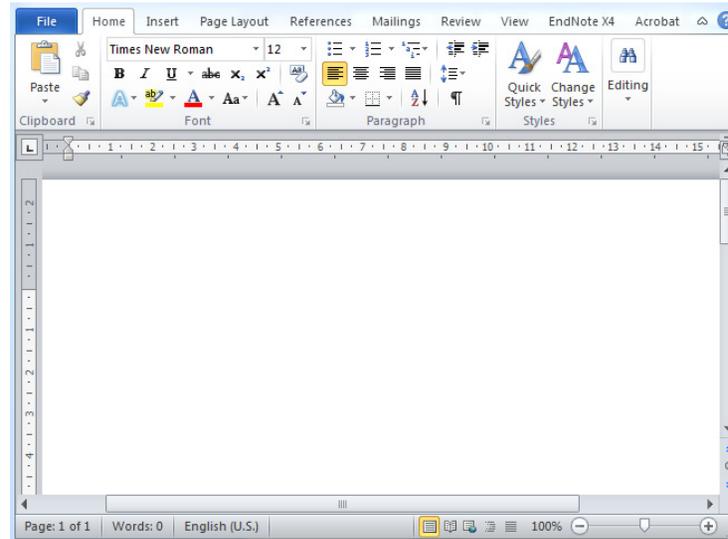


Karakteristik Berpikir Komputasi

- Mampu memberikan **pemecahan masalah** menggunakan **komputer** atau perangkat lain
- Mampu **mengorganisasi** dan **menganalisa data**
- Mampu melakukan representasi data melalui **abstraksi** dengan suatu **model** atau **simulasi**
- Mampu melakukan **otomatisasi solusi** melalui cara berpikir **algoritma** (sekumpulan langkah terurut)
- Mampu melakukan **identifikasi, analisa dan implementasi solusi** dengan berbagai kombinasi langkah/cara dan sumberdaya yang efisien dan efektif dengan
- Mampu melakukan **generalisasi solusi** untuk berbagai masalah berbeda



Pemecahan masalah dengan komputer



KU1072/P

Organisasi dan Analisa Data

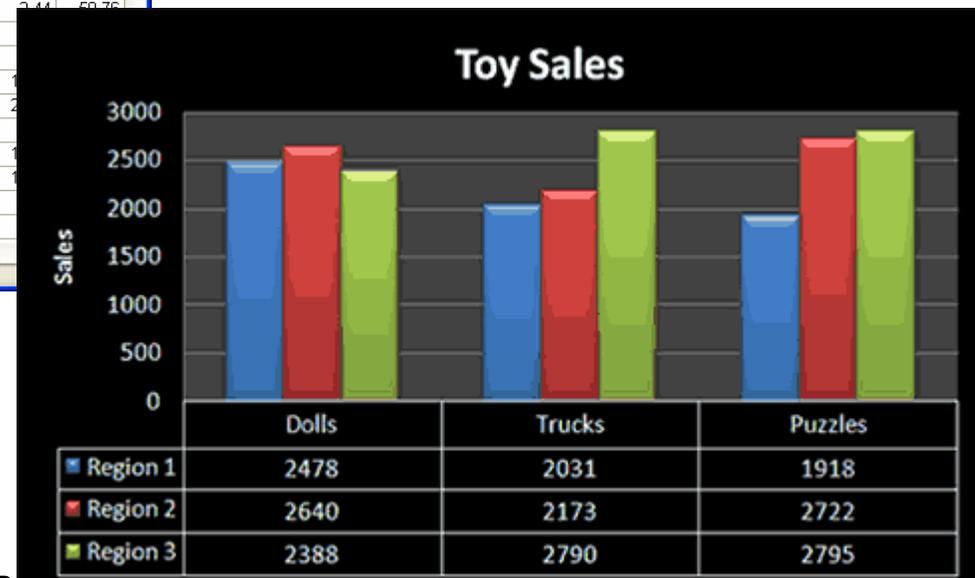
colchester2001.sav [DataSet1] - SPSS Data Editor

File Edit View Data Transform Analyze Graphs Utilities Window Help

1 : outputarea 22UGGE0001 Visible: 29 of 29

	outputarea	ward	v1	v2	v3	v4	v5	v6	v7	v8	v9
1	22UGGE0001	22UGG	9.59	14.38	35.27	19.18	5.14	51.85	2.78	5.50	65.17
2	22UGGE0002	22UGG	3.43	11.21	27.73	32.09	11.53	34.43	4.10	6.84	91.56
3	22UGGE0003	22UGG	5.56	11.40	29.53	26.02	8.19	48.25	2.63	15.57	79.24
4	22UGGE0004	22UGG	1.16	14.34	21.71	25.97	27.91	11.43	37.86	10.37	53.49
5	22UGGE0005	22UGG	3.75	14.33	31.74	24.57	14.68	64.06	26.56	2.54	69.28
6	22UGGE0006	22UGG	4.04	19.88	31.99	17.70	11.49	17.09	3.42	10.00	75.16
7	22UGGE0007	22UGG	6.33	11.71	26.58	25.00	15.19	13.22	22.31	12.28	66.46
8	22UGGE0008	22UGG	4.18	13.09	34.18	12.36	10.55	35.71	.00	7.50	39.27
9	22UGGE0009	22UGG	2.67	5.35	27.27	9.63	13.37	.00	65.00	25.71	35.64
10	22UGGE0010	22UGG	6.92	12.69	33.08	18.85	15.00	10.32	50.79	2.44	69.76
11	22UGGE0011	22UGG	7.96	12.74	24.84	25.16	14.97	37.10	25.81		
12	22UGGE0012	22UGG	9.60	18.36	40.96	19.21	4.52	48.82	.00		
13	22UGGE0013	22UGG	6.69	18.06	30.77	21.74	13.38	42.45	6.60		
14	22UGGE0014	22UGG	5.11	13.92	25.57	29.55	10.80	23.62	.00		
15	22UGGE0015	22UGG	5.83	20.86	26.99	16.26	19.33	31.71	2.44		
16	22UGGE0016	22UGG	7.35	19.17	34.19	23.32	4.79	35.78	.00		
17	22UGGE0017	22UGG	4.42	17.98	26.18	22.08	21.77	31.30	2.29		
18	22UGGE0018	22UGG	4.87	18.18	37.99	19.81	6.17	53.78	2.52		
19	22UGGE0019	22UGG	8.20	16.02	28.52	21.09	16.80	29.46	53.57		

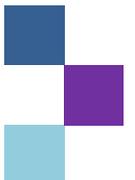
Data View Variable View / SPSS Processor is ready





Contoh Analisa Persoalan: Mengupas Kentang

- Bila ingin makan kentang, tentunya kita perlu memiliki kentang terlebih dahulu
- Jika belum ada, maka beli kentang dulu
- Jika sudah ada maka kentang perlu di kupas
- Setelah dikupas, kita harus memilih, apakah kita mau menggoreng kentang, merebus kentang atau membuat sup.



Mengupas Kentang

Kentang tersedia ?



Ya



Kupas Kentang

Mau dimasak apa?

Goreng ?



Rebus ?



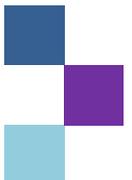
Sup ?



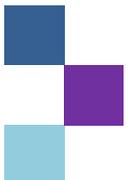
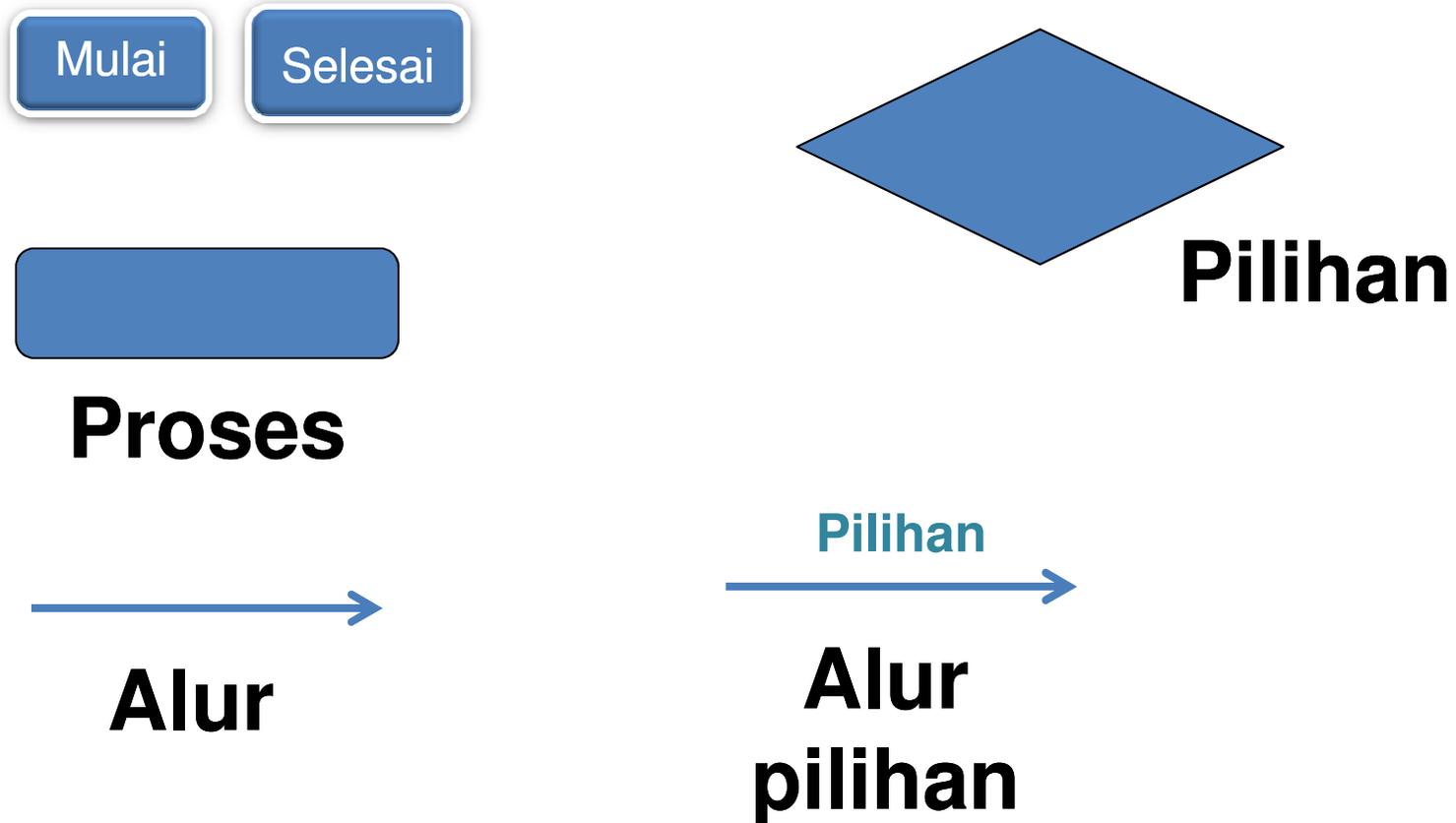
Tidak



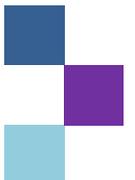
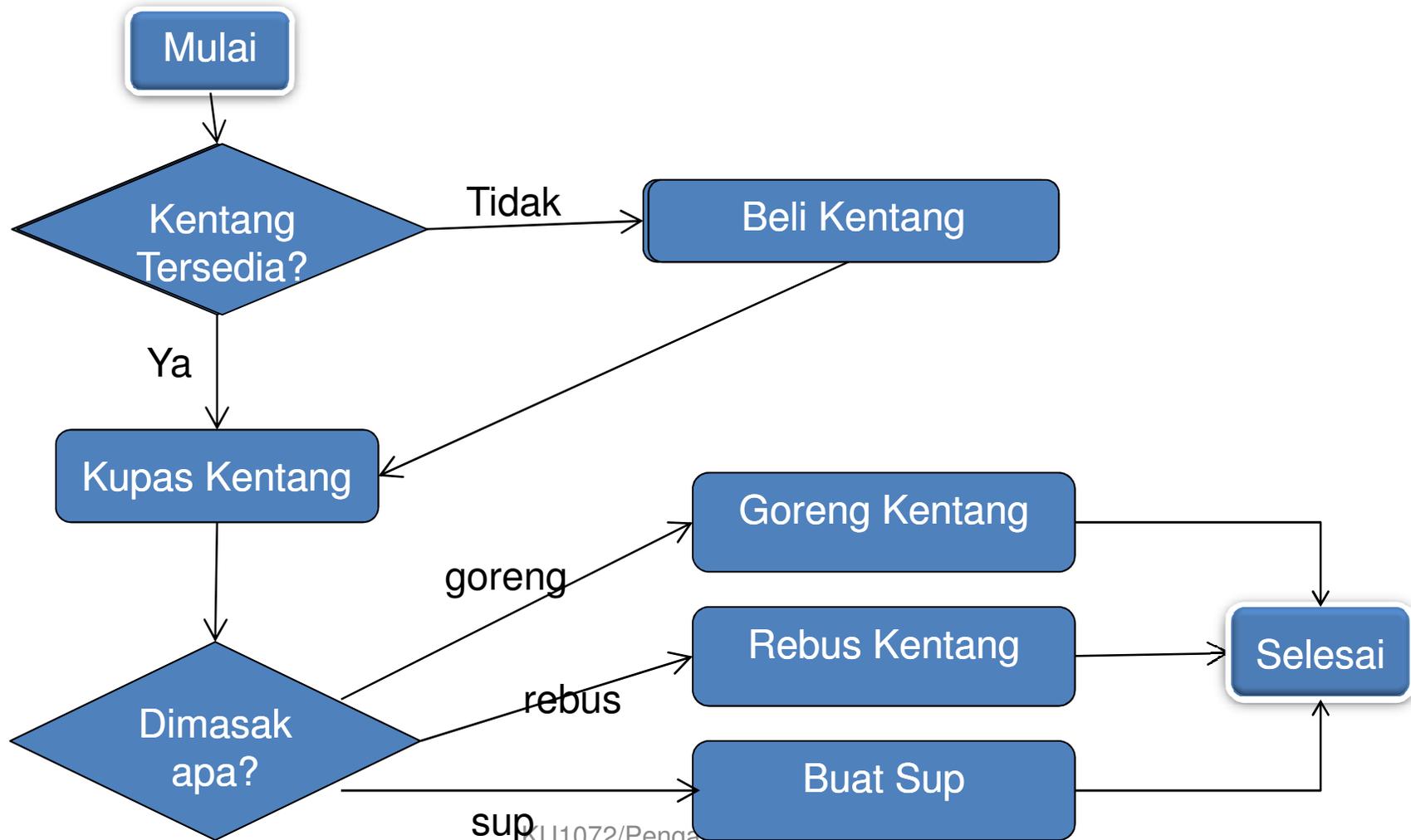
Beli kentang ?



Pengenalan Flowchart (Diagram Alir)



Flow Chart Mengupas Kentang





Contoh Analisa Persoalan: Jarum Jam dinding yang tidak tepat

- Bila jarum tidak bergerak, ganti battery
- Jika bergerak berarti battery masih hidup tinggal dilakukan perbaikan letak jarum jam
- Buatlah flowchartnya!

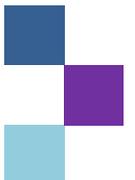


Diagram Alir (flow chart) Perbaiki Waktu di Jam dinding



Apa yang salah
dengan flowchart
ini??

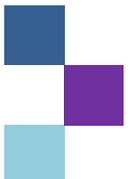


Diagram Alir (flow chart) Perbaiki Waktu di Jam dinding



Bagaimana jika ternyata setelah dua hari jam kembali tidak tepat?

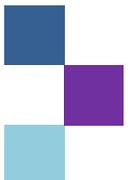
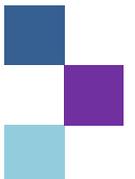
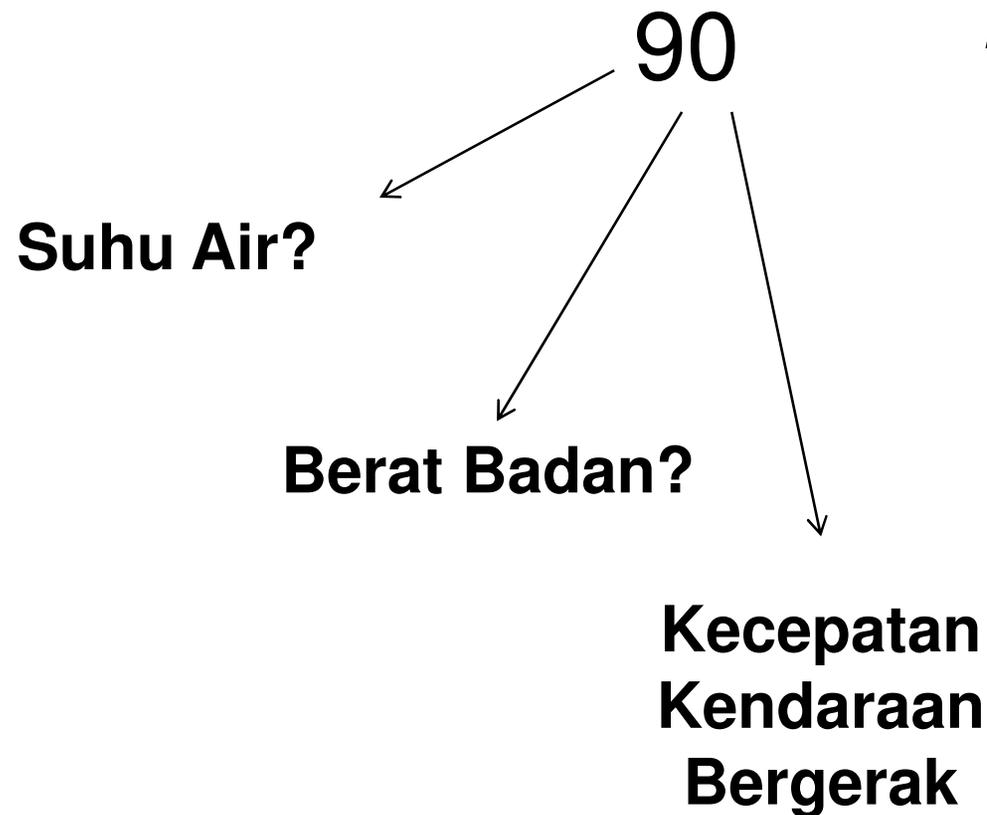


Diagram Alir (flow chart) Perbaiki Waktu di Jam dinding



Abstraksi Data

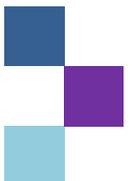
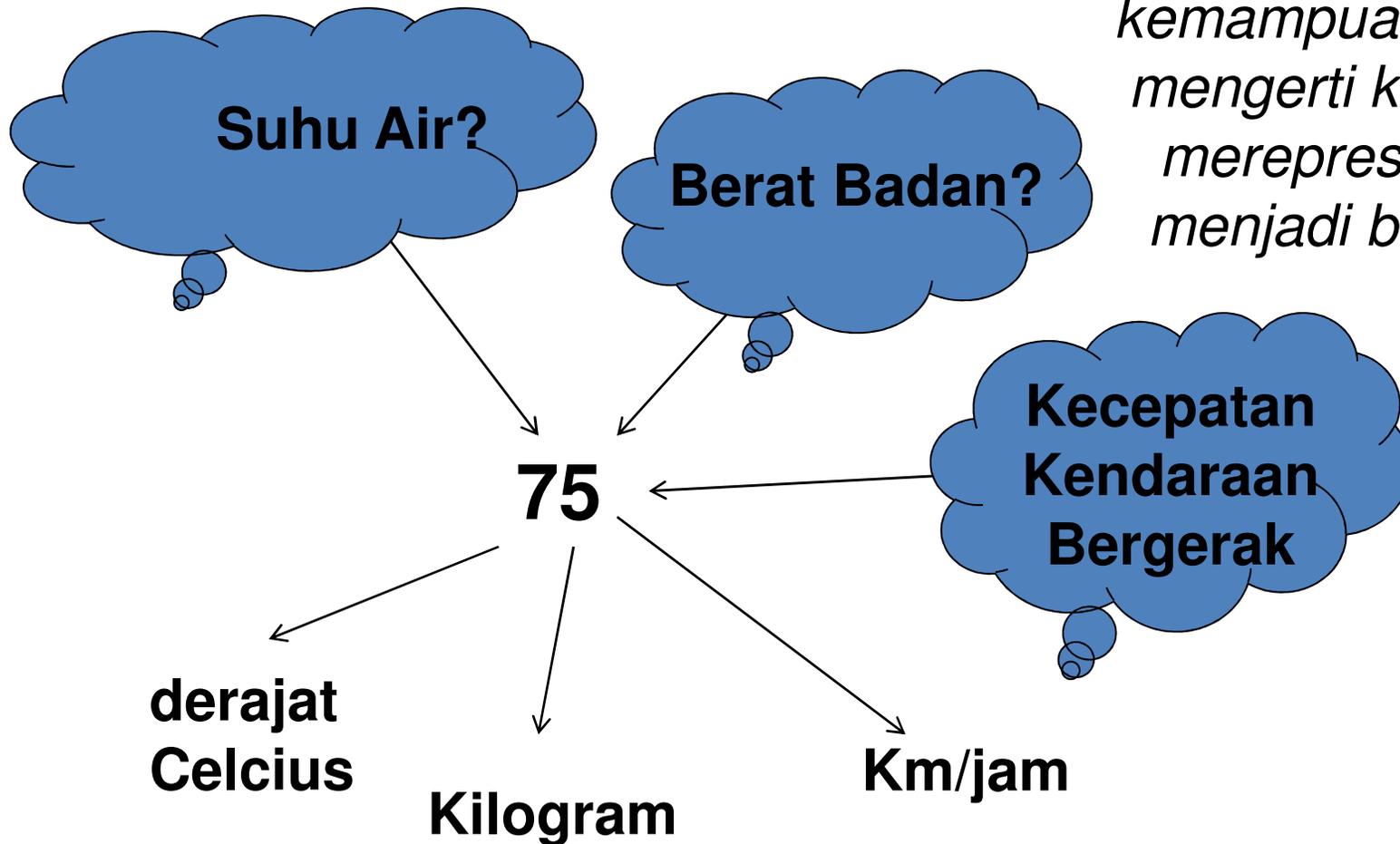


*kemampuan kita untuk
menginterpretasikan
suatu data dengan
konteks masalahnya*



Persoalan Abstraksi Data

kemampuan kita untuk mengerti konteks dan merepresentasikan menjadi bentuk lain.



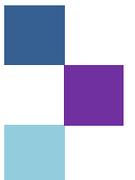


Data Mahasiswa dengan Data Penduduk

Struktur Data

NIM:
Nama
Kota Lahir
Tanggal Lahir
Nama Ayah
Nama Ibu

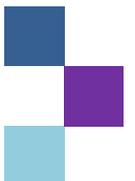
Nomor KTP:
Nama
Kota Lahir
Tanggal Lahir
Nama Ayah
Nama Ibu





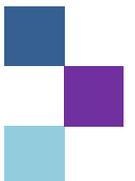
Abstraksi data dengan Model dan Simulasi

- Mampu melakukan **otomatisasi solusi** melalui cara berpikir algoritma (sekumpulan langkah terurut)
- Mampu melakukan **identifikasi, analisa dan implementasi solusi** dengan berbagai kombinasi langkah/cara dan sumberdaya yang efisien dan efektif dengan
- Mampu melakukan **generalisasi solusi** untuk berbagai masalah berbeda



Kemampuan minimum yang diharapkan

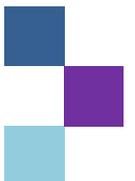
- Kemampuan melakukan dekomposisi masalah
 - Menganalisa resep suatu masakan
- Kemampuan mengenali pola
 - Misalnya pola harga barang yang naik kalau setiap lebaran
- Kemampuan menggeneralisir pola dan mengabstraksi pola
 - Abstraksi kalender kerja yang merepresentasikan apa yang dikerjakan dan kapan
- Kemampuan perancangan program
 - Instruksi untuk memasak dengan suatu resep
 - Instruksi membuka paket lemari knock down





Keuntungan berpikir komputasi

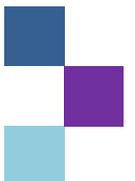
- **Percaya diri** dalam berhadapan dengan kompleksitas masalah yang lebih besar dan lebih sulit
- Toleransi terhadap **ambiguitas**
- Mampu berhubungan dengan masalah yang open-ended
- Mampu berkomunikasi dan bekerjasama untuk mencapai tujuan yang sama.





Berpikir Prosedural

- Sejumlah aksi dijalankan secara berurutan (sekuensial)
- Setiap aksi akan memberikan efek eksekusi tertentu
- Jika diikuti terus menerus, aksi-aksi ini harus selesai
 - Tidak bisa terus menerus



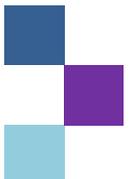


Pemrograman Prosedural

- Pemrograman Prosedural (Imperative)
 - Hasil eksekusi program berdasarkan hasil dekomposisi “aksional”.
 - Setiap aksi ini akan dijalankan secara berurutan (sekuensial)
- Pemrograman Non Prosedural
 - Tidak berdasarkan urutan sekuensial
 - Contoh:

- Pemrograman Deklaratif,
- Pemrograman Fungsional

**Tidak di
bahas di
kuliah ini**





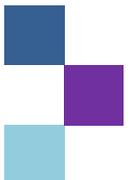
Program = Algoritma + Struktur Data





Program Menghitung Tabungan

- Masalah:
 - Tabungan di bank selalu bertambah setiap tahun
 - Bank memiliki bunga yang setiap tahun bertambah
 - BNI memiliki bunga tahunan 10% per tahun
 - Untuk uang 100 ribu, maka setelah satu tahun akan menjadi 110 ribu.
 - Buatlah program yang menghitung uang kita setelah satu tahun.
 - Program akan menanyakan uang kita saat ini, kemudian program akan menampilkan uang kita setelah satu tahun

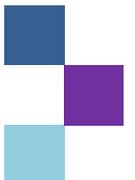




Contoh Eksekusi Program Menghitung Tabungan

Masukkan uang: 1000

Tahun Depan => 1100





Bagaimana Program itu dibuat?

KU1072/Pengantar Teknologi Informasi
B/Sem 1 2013-2014





Program Tabungan

input (NilaiRp)

NilaiRp \leftarrow NilaiRp + NilaiRp * 10%

output(NilaiRp)

atau

output (“Masukkan Uang: “)

input (NilaiRp)

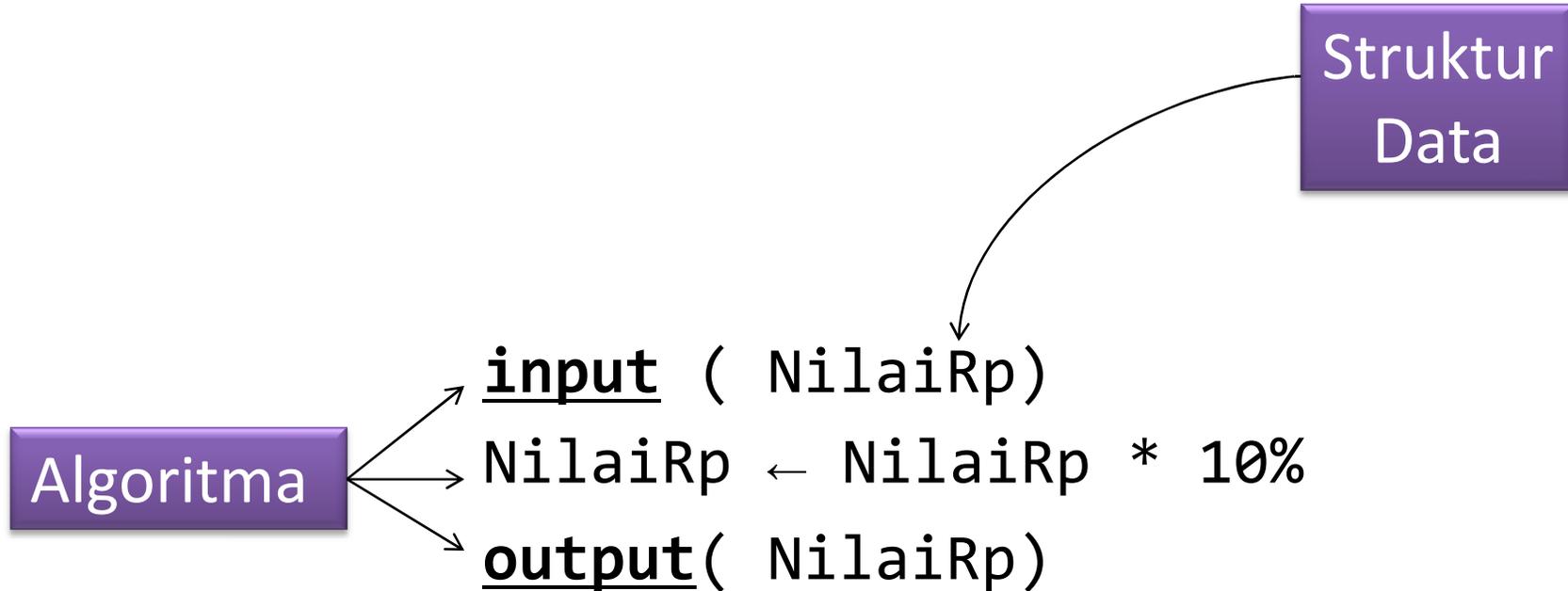
NilaiRp \leftarrow NilaiRp + NilaiRp * 10%

output(“Tahun Depan => “)

output(NilaiRp)



Program = Algoritma + Struktur Data





Kode Program Bahasa C++

input (NilaiRp)

NilaiRp ← NilaiRp * 10%

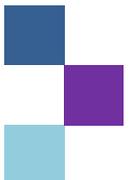
output(NilaiRp)



cin: Console Input
(diketikkan lewat
keyboard)

```
cin >> NilaiRp;  
NilaiRp = NilaiRp * 0.1;  
cout << NilaiRp;
```

cout: Console Output





Kode Program Bahasa Pascal

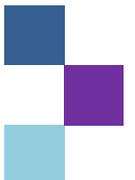
```
input ( NilaiRp)  
NilaiRp ← NilaiRp * 10%  
output( NilaiRp)
```



*Readln akan
membaca dari hasil
ketik di keyboard*

```
→ Readln(NilaiRp);  
NilaiRp = NilaiRp * 0.1;  
→ Writeln(NilaiRp);
```

*Writeln akan menulis
hasil di layar komputer*





Kode Program Bahasa Fortran

input (NilaiRp)

NilaiRp ← NilaiRp * 10%

output(NilaiRp)

Tanda "" mengindikasikan
keluaran/masukan
standard (keyboard/layar)*

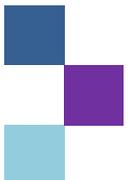
*Read akan membaca
dari hasil ketik di
keyboard*

Read *, NilaiRp

NilaiRp = NilaiRp * 0.1;

Print *, NilaiRp

*Print akan menulis
hasil di layar komputer*

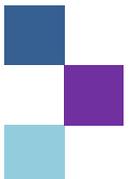
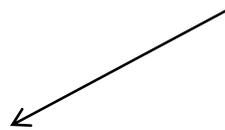




Kode C++

```
int main()  
{  
    int NilaiRp;  
  
    cin >> NilaiRp;  
    NilaiRp = NilaiRp + NilaiRp * 0.1;  
    cout << NilaiRp;  
    return 0;  
}
```

Pendefinisian Data





Kode Pascal

Program HitungUangDiBank **Pendefinisian Data**

var

 NilaiRp : integer;

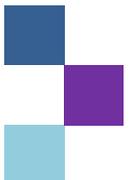
begin

 readln(NilaiRp);

 NilaiRp := NilaiRp + NilaiRp * 0.1;

 writeln(NilaiRp);

End.





Kode Fortran

Program HitungUangDiBank

Pendefinisian Data

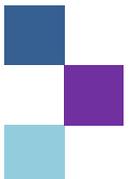
```
integer :: NilaiRp
```

```
read *, NilaiRp
```

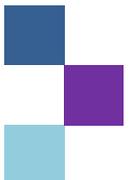
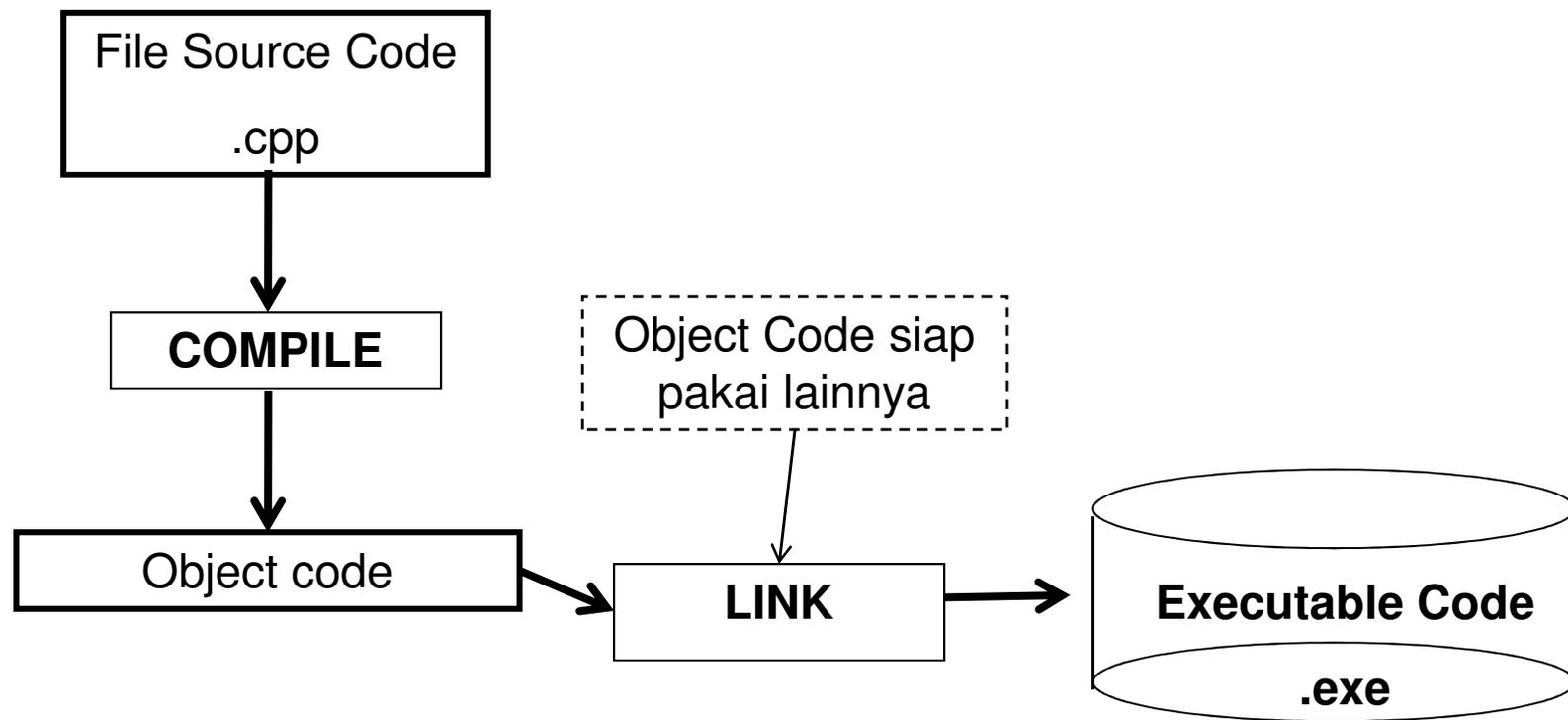
```
NilaiRp := NilaiRp + NilaiRp * 0.1
```

```
print *, NilaiRp
```

```
end program HitungUangDiBank
```



Edit, Kompilasi (Compile) dan Eksekusi



Contoh Edit, Kompilasi (Compile) dan Eksekusi



```
main.cpp [ku1071-bungatabungan] - Code::Blocks 10.05
File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help
Management
Projects Symbols
Workspace
ku1071-bungatabungan
Sources
main.cpp
C:\Users\bayu\Documents\ProjectC\ku1071-bungatabungan\bin\Debug\ku1071-bungatabungan.exe
1000
1100
Process returned 0 (0x0) execution time : 5.604 s
Press any key to continue.
```

KU1072/Pengantar Teknologi Informasi
B/Semester 1 2013-2014



Program HitungLuasLingkaran (C++)

```
int main()
{
    /* Kamus */
    float JariJari;
    float Luas;

    /* Algoritma */
    cin >> JariJari;
    Luas = 3.14 * JariJari * JariJari;
    cout << Luas;
}
```





Program HitungLuasLingkaran (Pascal)

```
Program HitungLuasLingkaran;  
{ Kamus }  
var  
    JariJari: real;  
    Luas     : real;  
begin  
    {Algoritma }  
    readln(JariJari);  
    Luas := 3.14 * JariJari * JariJari;  
    writeln(Luas);  
    readln;  
end.
```





Program HitungLuasLingkaran (Fortran)

Program HitungLuasLingkaran

! Kamus

```
real :: JariJari
```

```
real :: Luas
```

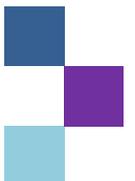
! algoritma

```
read *, JariJari
```

```
Luas = 3.14 * JariJari * JariJari
```

```
print *, Luas
```

end program HitungLuasLingkaran





Program HitungLuasLingkaran (Fortran) – versi 2

Program HitungLuasLingkaran

! Kamus

```
real :: JariJari
```

```
real :: Luas
```

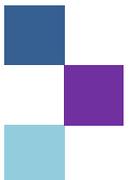
! algoritma

```
read *, JariJari
```

```
Luas = 3.14 * JariJari ** 2
```

```
print *, Luas
```

end program HitungLuasLingkaran





Terima Kasih

KU1072/Pengantar Teknologi Informasi
B/Sem 1 2013-2014

