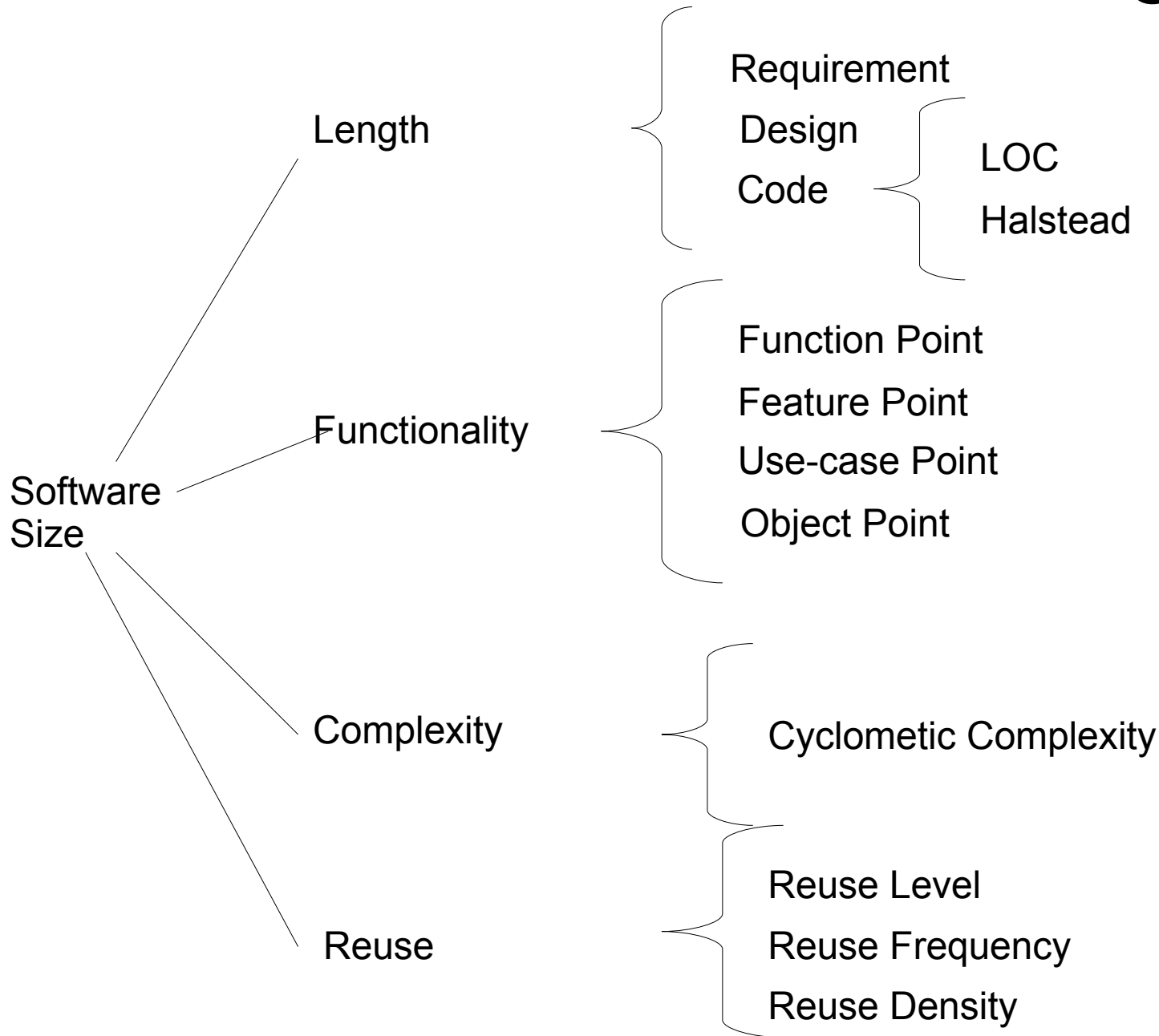


Mengukur Atribut Produk Internal: UKURAN

Ukuran Software

- Atribut-atribut Produk Internal mendeskripsikan sebuah produk software dalam suatu cara yang tergantung hanya pada produk itu sendiri.
- **Ukuran** merupakan salah satu atribut produk software yang dapat dihitung secara statistik.
- Pengukuran ukuran harus mencerminkan *effort*, *cost* dan *productivity*.

Software Size Metric: Ringkasan



Panjang Software

- Panjang adalah ukuran fisik suatu produk
- Ada 3 produk pengembangan software:
 - spesifikasi
 - design
 - code
- Panjang spesifikasi mengindikasikan panjang design, yang dapat digunakan untuk memprediksi panjang kode.

code counting checklist

Definition name: Physical Source Lines of Code Date: 8/7/92
(basic definition) Originator: SEI

Measurement unit:		Physical source lines	<input checked="" type="checkbox"/>	Logical source statements	<input type="checkbox"/>		
Statement type	Definition	<input checked="" type="checkbox"/>	Data array	<input type="checkbox"/>		Includes	Excludes
	When a line or statement contains more than one type, classify it as the type with the highest precedence.						
	Order of precedence ->						
1	Executable				1	✓	
2	Nonexecutable						
3	Declarations				2	✓	
4	Compiler directives				3	✓	
5	Comments						
6	On their own lines				4		✓
7	On lines with source code				5		✓
8	Banners and nonblank spaces				6		✓
9	Blank (empty) comments				7		✓
10	Blank lines				8		✓
11							
12							
How produced	Definition	<input checked="" type="checkbox"/>	Data array	<input type="checkbox"/>		Includes	Excludes
1	Programmed					✓	
2	Generated with source code generators					✓	
3	Converted with automated translators					✓	
4	Copied or reused without change					✓	
5	Modified					✓	
6	Removed						✓
7							
8							
Origin	Definition	<input checked="" type="checkbox"/>	Data array	<input type="checkbox"/>		Includes	Excludes
1	New work: no prior existence					✓	
2	Prior work: taken or adapted from						
3	A previous version, build, or release					✓	
4	Commercial, off-the-shelf software (COTS), other than libraries					✓	
5	Government furnished software (GFS), other than reuse libraries					✓	
6	Another product					✓	
7	A vendor-supplied language support library (unmodified)						✓
8	A vendor-supplied operating system or utility (unmodified)						✓
9	A local or modified language support library or operating system					✓	
10	Other commercial library					✓	
11	A reuse library (software designed for reuse)					✓	
12	Other software component or library					✓	
13							

LOC

- eLOC (effective LOC) adalah ukuran dari semua baris yang tidak komentar, kosong atau *brace* atau kurung.
- ILOC (logical LOC) : LOC yang mewakili perintah dalam program.

Contoh LOC

Source code line	LOC	eLOC	lLOC	Comment	Blank
if (x<10) // test range	x	x		x	
{	x				
// update y coordinate				x	
y = x + 1;	x	x	x		x
}	x				

Gambaran SLOC

Year	Operating System	SLOC (Million)
1993	Windows NT 3.1	4-5 ^[1]
1994	Windows NT 3.5	7-8 ^[1]
1996	Windows NT 4.0	11-12 ^[1]
2000	Windows 2000	more than 29 ^[1]
2001	Windows XP	40 ^[1]
2003	Windows Server 2003	50 ^[1]

Operating System	SLOC (Million)
Debian 2.2	55-59 ^{[2][3]}
Debian 3.0	104 ^[3]
Debian 3.1	215 ^[3]
Debian 4.0	283 ^[3]
Debian 5.0	324 ^[3]
OpenSolaris	9.7
FreeBSD	8.8
Mac OS X 10.4	86 ^[4]
Linux kernel 2.6.0	5.2
Linux kernel 2.6.29	11.0
Linux kernel 2.6.32	12.6 ^[5]

Tugas #1

- Cari keuntungan dan kerugian menggunakan LOC fisik untuk pengukuran panjang software!
- Tulisan aturan perhitungan LOC untuk bahasa C! (Ambil dan sebutkan dari sumber yang Anda gunakan).

Metode Halstead

- Teori Maurice Halstead (1971-1979)
 - sebuah program P adalah kumpulan token, tersusun dari 2 elemen dasar: **operan** dan **operator**
 - Operan: variabel, konstanta, alamat
 - operator: operasi dalam bahasa pemrograman
 - for, if...else..., + - =, main(), goto

Metode Halstead

- jumlah operator yang berbeda dalam program (μ_1)
- jumlah operan yang berbeda dalam program (μ_2)
- total jumlah kemunculan operator dalam program (N_1)
- total jumlah kemunculan operan dalam program (N_2)
- program vocabulary (μ)
 - $\mu = \mu_1 + \mu_2$

Metode Halstead

- Panjang program:
 - $N = N_1 + N_2$
- Volume Program:
 - $V = N \log_2 \mu$
- Level Program:
 - $L = V^*/V$ atau $\hat{L} = \frac{1}{D} = \frac{2}{\mu_1} \times \frac{\mu_2}{N_2}$
- V^* adalah minimal ukuran implementasi
- D adalah level kesulitan = $1/L$

Metode Halstead

- Perkiraan panjang program:
 - $\hat{N} = \mu_1 \times \log_2 \mu_1 + \mu_2 \times \log_2 \mu_2.$
- Effort untuk menghasilkan program:
 - $$E = \frac{V}{\hat{L}} = \frac{\mu_1 N_2 N \log_2 \mu}{2\mu_2}$$
- Waktu yang diperlukan :
 - $T = E / \beta$
 - β antara 5 – 20
 - Halstead mengklaim β bernilai 18

Metode Halstead

- Bug yang masih ada: bug yang masih ada saat delivery time:
 - $B = E^{2/3} / 3000$
- Kesimpulan:
 - program besar perlu waktu lebih banyak untuk pengembangan demikian juga dengan bug remain.

Contoh

```
#include<stdio.h>
main()
{
int a ;
scanf ("%d", &a);
if ( a >= 10 )
if ( a < 20 ) printf ("10 < a< 20 %d\n" , a);
else      printf ("a >= 20 %d\n" , a);
else      printf ("a <= 10 %d\n" , a);
}
```

Contoh

Opeartor	Kemunculan	Operator	Kemunculan
#		<=	
include		\n	
stdio.h		printf	
<...>		<	
main		>=	
(...)		if ... else	
{...}		&	
int		,	
;		%d	
scanf		" ... "	
$\mu 1 =$		N1 =	

Contoh

Operand	Kemunculan
a	
10	
20	

Panjang Program ?

Perkiraan Panjang Program ?

Contoh

Opeartor	Kemunculan	Operator	Kemunculan
#	1	<=	1
include	1	\n	3
stdio.h	1	printf	3
<...>	1	<	3
main	1	>=	2
(...)	7	if ... else	2
{...}	1	&	1
int	1	,	4
;	5	%d	4
scanf	1	" ... "	4
$\mu 1 = 20$		N1 = 47	

Contoh

Operand	Kemunculan
a	7
10	1
20	1

$\mu_2 = 3$	$N_2 = 9$
$\mu_1 = 20$	$N_1 = 47$

Panjang Program ?

Perkiraan Panjang Program ?

Perkiraan LOC

- Langkah pertama dalam perkiraan berbasis LOC adalah memperkirakan jumlah baris kode program pada akhir proyek.
- Dapat dilakukan berdasar
 - pengalaman,
 - ukuran proyek sebelumnya,
 - ukuran dari kompetitor, atau
 - membagi proyek ke dalam bagian-bagian yang lebih kecil dan kemudian memperkirakan ukuran setiap bagian yang lebih kecil tersebut.

Perkiraan LOC

- Pendekatan baku, untuk setiap bagian (P_i), adalah memperkirakan
 - ukuran maksimum yang mungkin (\max_i),
 - ukuran minimum yang mungkin (\min_i),
 - Tebakan ukuran terbaik (best_i).
- Perkiraan untuk seluruh proyek adalah $1/6$ dari jumlah maksimum, minimum, dan $4 * \text{best}$.
 - Simpangan baku (S) =
$$\sqrt{\sum_{i=1}^n (\max_i - \min_i)^2}$$
 - Perkiraan tiap $P_i = (\min_i + \max_i + 4 * \text{best}_i) / 6$

Part	Min Size	Best Guest	Max Size
1	20	30	50
2	10	15	25
3	25	30	45
4	30	35	40
5	15	20	25
6	10	12	14
7	20	22	25

Contoh

$$p1(20 + 4 * 30 + 50)/6 = 190/6 = 31.6$$

$$p2(10 + 4 * 15 + 25)/6 = 95/6 = 15.8$$

$$p3(25 + 4 * 30 + 45)/6 = 190/6 = 31.6$$

$$p4(30 + 4 * 35 + 40)/6 = 220/6 = 36.7$$

$$p5(15 + 4 * 20 + 25)/6 = 120/6 = 20$$

$$p6(10 + 4 * 12 + 14)/6 = 72/6 = 12$$

$$p7(20 + 4 * 22 + 25)/6 = 133/6 = 22.17$$

- Total LOC = 170.07

$$\begin{aligned}
 \text{Standard deviation} &= ((50 - 20)^2 + (25 - 10)^2 + (45 - 25)^2 \\
 &+ (40 - 30)^2 + (25 - 15)^2 + (14 - 10)^2 + (25 - 20)^2)^{.5} \\
 &= (900 + 225 + 400 + 100 + 100 + 16 + 25)^{.5} \\
 &= 1766^{.5} = 42.03
 \end{aligned}$$

```
#include "stdio.h"
// deklarasi fungsi
void bubbleSort(int numbers[], int array_size);
// fungsi utama
//
void main() {
    int data[] = {4,2,6,9,3,1};
    int i;

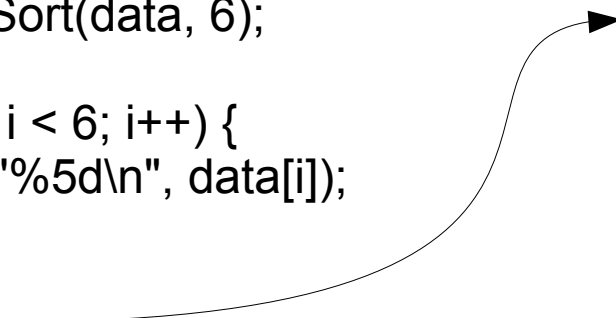
    bubbleSort(data, 6);

    for(i=0; i < 6; i++) {
        printf("%5d\n", data[i]);
    }
}
```

Tugas #2

```
void bubbleSort(int numbers[], int array_size)
{
    int i, j, temp;

    for (i = (array_size - 1); i >= 0; i--)
    {
        for (j = 1; j <= i; j++)
        {
            if (numbers[j-1] > numbers[j])
            {
                temp = numbers[j-1];
                numbers[j-1] = numbers[j];
                numbers[j] = temp;
            }
        }
    }
}
```



Tugas #2

- Hitung eLOC dan LLOC untuk program di slide 18!
- Dengan menggunakan metode Halstead, hitung panjang program dan perkiraan panjang program!

Perkiraan LOC

- Langkah pertama dalam perkiraan berbasis LOC adalah memperkirakan jumlah baris kode program pada akhir proyek.
- Dapat dilakukan berdasar
 - pengalaman,
 - ukuran proyek sebelumnya,
 - ukuran dari kompetitor, atau
 - membagi proyek ke dalam bagian-bagian yang lebih kecil dan kemudian memperkirakan ukuran setiap bagian yang lebih kecil tersebut.

Perkiraan LOC

- Pendekatan baku, untuk setiap bagian (P_i), adalah memperkirakan
 - ukuran maksimum yang mungkin (\max_i),
 - ukuran minimum yang mungkin (\min_i),
 - Tebakan ukuran terbaik (best_i).
- Perkiraan untuk seluruh proyek adalah $1/6$ dari jumlah maksimum, minimum, dan $4 * \text{best}$.
 - Simpangan baku (S) =
$$\sqrt{\sum_{i=1}^n (\max_i - \min_i)^2}$$
 - Perkiraan tiap $P_i = (\min_i + \max_i + 4 * \text{best}_i) / 6$

Part	Min Size	Best Guest	Max Size
1	20	30	50
2	10	15	25
3	25	30	45
4	30	35	40
5	15	20	25
6	10	12	14
7	20	22	25

Contoh

$$p1(20 + 4 * 30 + 50)/6 = 190/6 = 31.6$$

$$p2(10 + 4 * 15 + 25)/6 = 95/6 = 15.8$$

$$p3(25 + 4 * 30 + 45)/6 = 190/6 = 31.6$$

$$p4(30 + 4 * 35 + 40)/6 = 220/6 = 36.7$$

$$p5(15 + 4 * 20 + 25)/6 = 120/6 = 20$$

$$p6(10 + 4 * 12 + 14)/6 = 72/6 = 12$$

$$p7(20 + 4 * 22 + 25)/6 = 133/6 = 22.17$$

- Total LOC = 170.07

$$\begin{aligned}
 \text{Standard deviation} &= ((50 - 20)^2 + (25 - 10)^2 + (45 - 25)^2 \\
 &+ (40 - 30)^2 + (25 - 15)^2 + (14 - 10)^2 + (25 - 20)^2)^{.5} \\
 &= (900 + 225 + 400 + 100 + 100 + 16 + 25)^{.5} \\
 &= 1766^{.5} = 42.03
 \end{aligned}$$

Tugas #3

- Buat perkiraan ukuran aplikasi web yang ditulis dengan bahasa PHP yang menyediakan fungsi:
 - galeri foto berbasis web
 - email client berbasis web
 - web store
- Yang perlu ditekankan:
 - aplikasi web merupakan aplikasi yang 80% adalah php (tidak mengandalkan javascript untuk interaksi)