

*Kontrak  
Perkuliahan*

*Review  
Rekayasa  
Perangkat Lunak*

*Manajemen  
Kualitas*

*Strategi &  
Teknik Testing*

*Implementasi  
Sistem*

*Suplement*

# **Teknik Testing**

Dr. Karmilasari

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

Teknik Testing

## Pengujian Aplikasi **KONVENSIONAL**

# Testing dan Implementasi Sistem

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Teknik Testing

- Filosofi pengujian yang harus ditanamkan pada diri pengembang **“buang jauh-jauh anggapan benar dari PL yang telah dikembangkannya dan berusaha untuk merancang suatu test case untuk menghancurkan PL tersebut”**
- Tujuan pengujian aplikasi konvensional : merancang **serangkaian test case yang mampu menyingkapkan kesalahan-kesalahan.**
- Teknik pengujian haruslah :
  - Memperlihatkan logika internal dan antarmuka dari setiap komponen PL
  - Memperlihatkan ranah masukan dan keluaran dari program untuk menyingkapkan kesalahan-kesalahan dalam fungsi, perilaku dan kinerja program.
- Yang melakukan pengujian :
  - Tahap awal pengujian : rekayasawan PL
  - Saat proses pengujian berlangsung : spesialis pengujian (tester)

# Testing dan Implementasi Sistem

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Teknik Testing

- Secara umum langkah-langkah pengujian dipandang dari dua sudut pandang berbeda :
  - Logika program internal diuji dengan teknik perancangan test case **White Box** (kotak putih)
  - Kebutuhan PL uji menggunakan teknik perancangan test case **Black Box** (kotak hitam)
- Use case membantu perancangan pengujian, yaitu membantu menyingkapkan kesalahan-kesalahan di tingkat validasi PL.
- Pada setiap kasus tujuannya adalah menemukan sebanyak mungkin kesalahan dengan sesedikit mungkin waktu dan usaha.
- Test case dirancang untuk menguji logika internal, antarmuka, kolaborasi komponen-komponen dan menguji kebutuhan eksternal yang telah dirancang dan didokumentasikan, hasil-hasil yang diharapkan yang telah ditetapkan dan hasil-hasil aktual yang telah dicatat.

## Dasar-dasar Pengujian

Kemampuan sebuah program komputer, untuk diuji (testability) meliputi :

- **Operability** (kemampuan untuk bisa dioperasikan), “semakin baik kinerjanya, semakin efisien PL untuk bisa diuji”
- **Observability** (kemampuan untuk bisa diobservasi), “apa yang dilihat adalah apa yang diuji”
- **Controllability** (kemampuan untuk dapat dikontrol), “ semakin baik PL dikontrol, semakin pengujian dapat diotomatisasi dan dioptimalkan”
- **Decomposability** (kemampuan untuk dapat disusun), “dengan mengontrol ruang lingkup pengujian, maka pengisolasian masalah dapat cepat dilakukan dan pengujian ulang dapat dilakukan dengan lebih cerdas”

# Testing dan Implementasi Sistem

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Teknik Testing

- **Simplicity** (kesederhanaan), “semakin sedikit yang diuji, semakin cepat dapat mengujinya”
- **Stability** (stabilitas), “ makin sedikit perubahan, semakin sedikit gangguan untuk pengujian”
- **Understability** (kemampuan untuk dapat dipahami), “semakin banyak informasi yang dimiliki semakin cerdas pengujianya”

# Testing dan Implementasi Sistem

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Teknik Testing

Karakteristik Pengujian :

- Pengujian yang baik memiliki probabilitas tinggi untuk menemukan kesalahan
- Pengujian yang baik tidak berulang-ulang
- Pengujian yang baik harus menjadi “bibit terbaik”
- Pengujian yang baik harus tidak terlalu sederhana atau terlalu rumit

## PENGUJIAN KOTAK PUTIH/WHITE BOX

- Disebut juga Pengujian Kotak Kaca / Glass-Box testing
- Pengujian dari sudut pandang pengembang
- Didasarkan pada pemeriksaan teliti terhadap detail prosedur. Jalur logis di seluruh PL dan kolaborasi antar komponen diuji dengan menguji serangkaian kondisi atau loop spesifik.
- Filosofi perancangan test case menggunakan struktur kontrol
- Test case yang diharapkan :
  - Menjamin semua jalur independen di dalam modul telah dieksekusi sedikitnya satu kali
  - Melaksanakan semua keputusan logis pada sisi benar dan salah
  - Melaksanakan semua loop pada batas merka dan dalam batas-batas operasionalnya
  - Melakukan struktur data internal dan memastikan kesahihannya

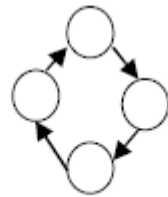


## Pengujian Jalur Dasar / Basis Path Testing

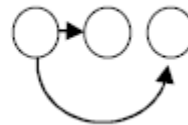
- Notasi Grafik Alir (Flow graph), menggambarkan arus kontrol logis dengan menggunakan notasi grafis.



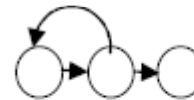
Sequence



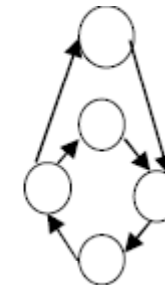
If



While



Until



Case

# Testing dan Implementasi Sistem

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

Teknik Testing

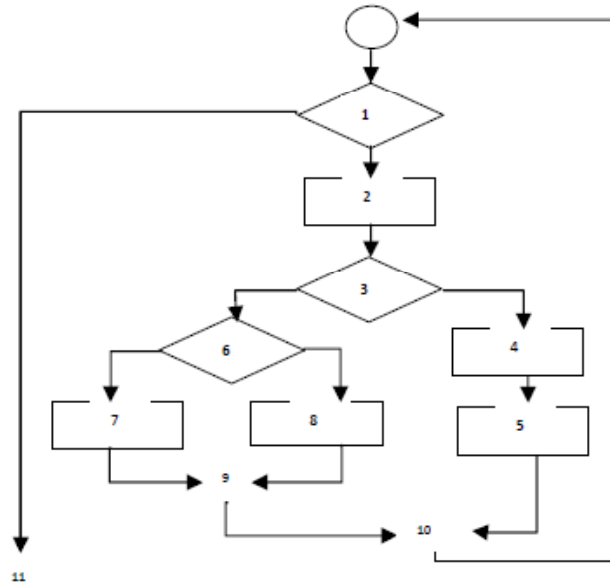
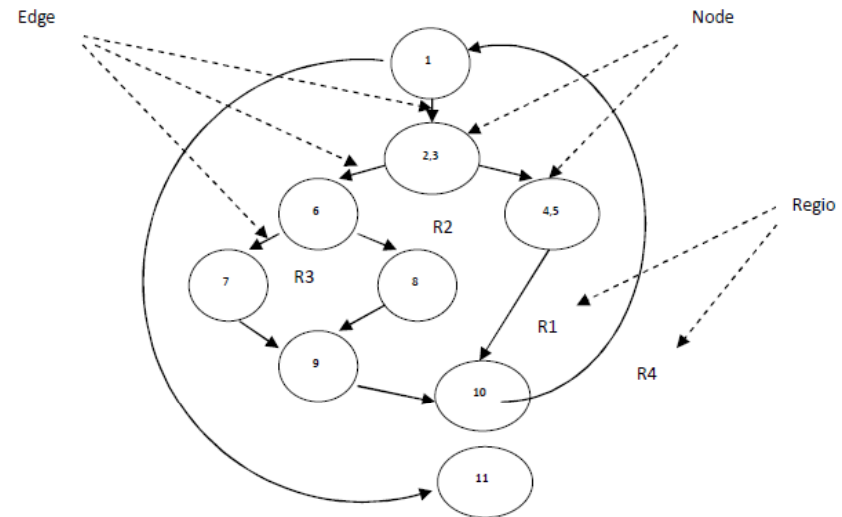
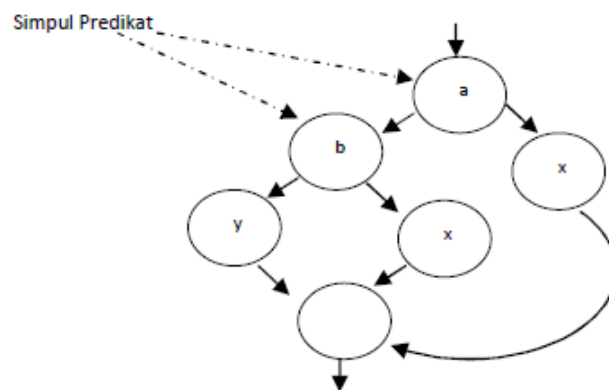


Diagram Alir



Grafik Alir



Logika Gabungan

```

    IF a OR b
    Then procedure x
    Else procedure y
    ENDIF
  
```

## Jalur Program Independen

- Adalah setiap jalur yang melalui program setidaknya satu kumpulan pernyataan pemrosesan atau kondisi baru.
- Bila dinyatakan dalam grafik alir, jalur independen harus bergerak sepanjang setidaknya satu edge yang belum dilintasi sebelum jalur tersebut didefinisikan

- Contoh, basis set :

Path 1 : 1-11

Path 2 : 1-2-3-4-5-10-1-11

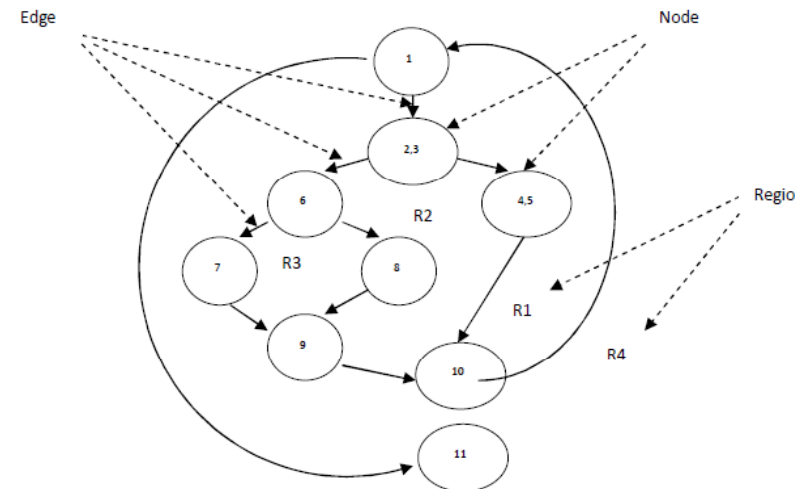
Path 3 : 1-2-3-6-8-9-10-1-11

Path 4 : 1-2-3-6-7-9-10-1-11

Path berikut bukan path independen,

karena merupakan kombinasi dari path di atas :

1-2-3-4-5-10-1-2-3-6-8-9-10-1-11



Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Kompleksitas Siklomatik

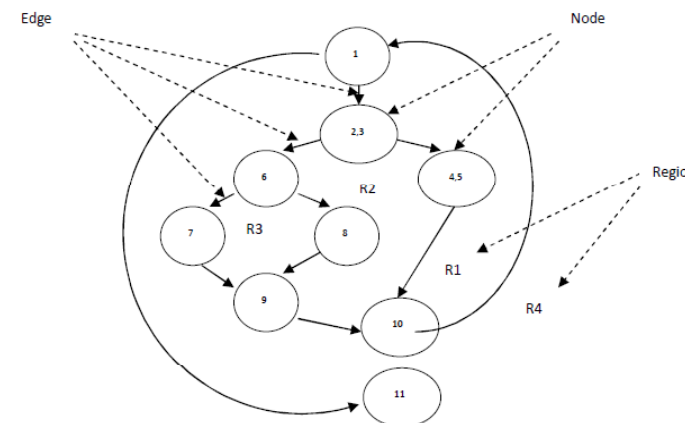
- Adalah metrik PL yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program
- Bila digunakan dalam konteks metode pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program dan menyediakan batas atas untuk jumlah pengujian yang harus dilakukan guna memastikan bahwa semua pernyataan telah dieksekusi minimal sekali.

## Perhitungan Kompleksitas Siklomatik :

1. Jumlah daerah-daerah grafik alir yang berhubungan dengan kompleksitas siklomatik
2. Kompleksitas siklomatik  $V(G)$  untuk grafik alir  $G$  didefinisikan sebagai  $V(G) = E - N + 2$
3. Kompleksitas siklomatik  $V(G)$  untuk grafik aliran  $G$  juga didefinisikan sebagai  $V(G) = P + 1$ , dimana  $P$  adalah jumlah node predikat yang terdapat dalam grafik alir  $G$ .

Contoh :

1. Grafik aliran memiliki 4 region
2.  $V(G) = 11 \text{ tepi} - 9 \text{ node} + 2 = 4$
3.  $V(G) = 3 \text{ node predikat} + 1 = 4$
4. Kompleksitas grafik alir adalah 4



## Menghasilkan Test Case

- Metode pengujian jalur-jalur dasar dapat diterapkan untuk perancangan prosedural atau kode program
- Langkah-langkahnya :
  1. Menggunakan perancangan atau kode sebagai sebuah dasar, buatlah gambar grafik alir
  2. Tentukan kompleksitas siklomatik dari aliran grafik yang dihasilkan
  3. Tentukan sebuah basis set dari lalur independen linier
  4. Menyiapkan test case yang akan memaksa pelaksanaan setiap jalur di basis set

# Testing dan Implementasi Sistem

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

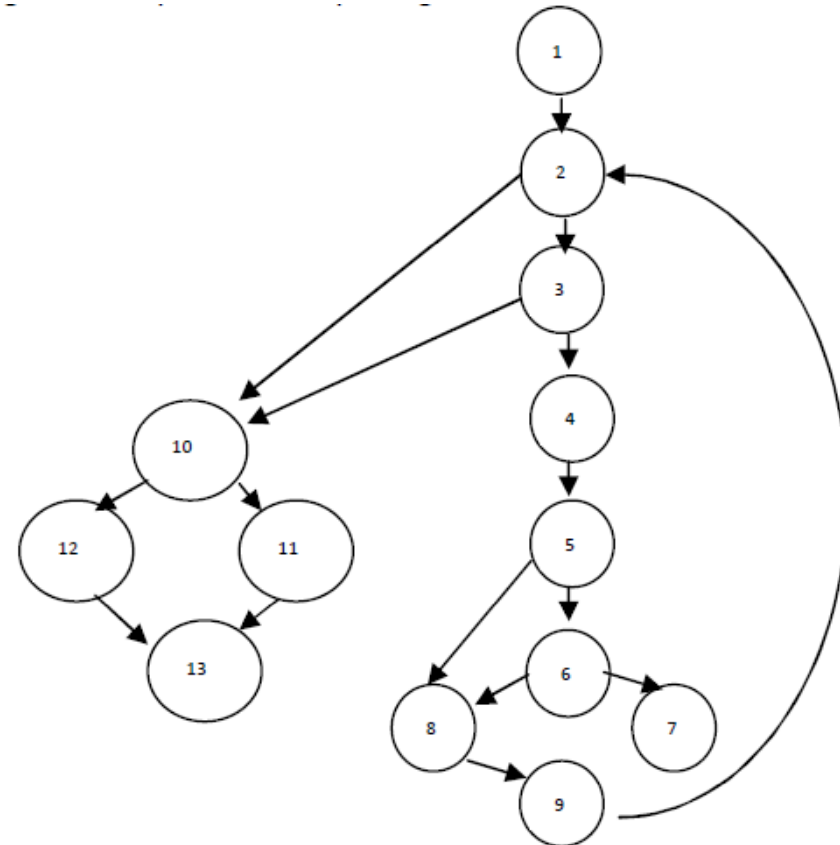
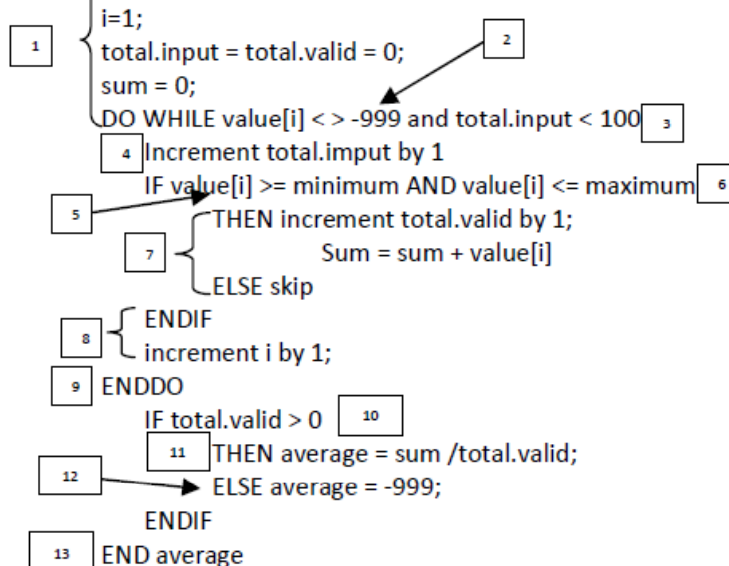
## Contoh Procedure Design Language & Grafik Alir :

PROCEDURE average

\*This procedure compute the average of 100 or fewer numbers that lie between bounding values; it also compute the sum and teh total number valid

INTERFACE RETURNS average, total.input, total.valid;  
INTERFACE ACCEPTS value, minimum, maximum;

TYPE value[1:11] IS SCALAR ARRAY;  
TYPE average, total.input, total.valid;  
minimum, maximum, sum IS SCALAR  
TYPE i IS INTEGER



Teknik Testing

# Testing dan Implementasi Sistem

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Teknik Testing

- Kompleksitas siklomatik untuk contoh di atas :

$$V(G) = 5 \text{ region}$$

$$V(G) = 16 \text{ edge} - 13 \text{ simpul} + 2 = 5$$

$$V(G) = 4 \text{ simpul predikat} + 1 = 5$$

- Basis set dari jalur independen untuk contoh di atas :

Jalur 1 : 1-2-10-11-13

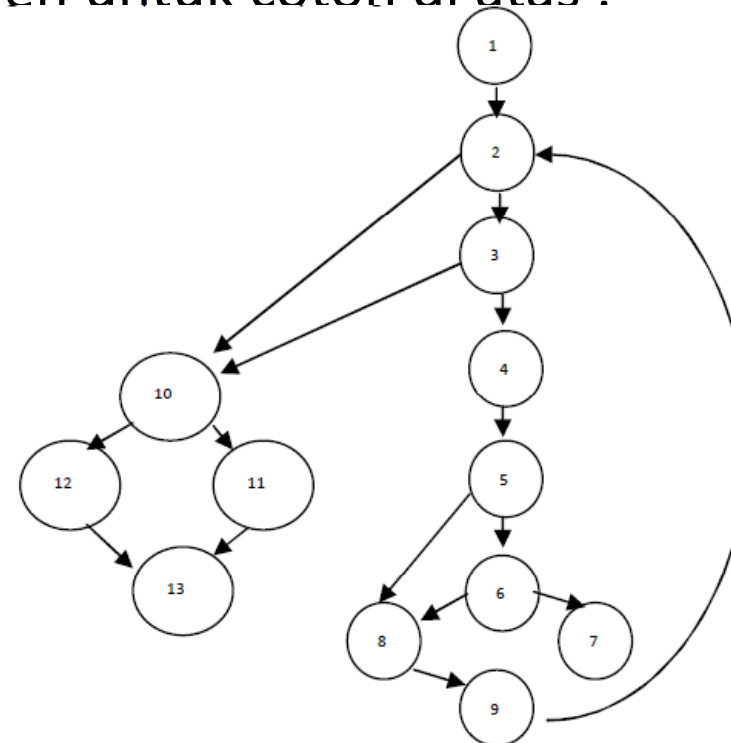
Jalur 2 : 1-2-10-12-13

Jalur 3 : 1-2-3-10-11-13

Jalur 4 : 1-2-3-4-5-8-9-2

Jalur 5 : 1-2-3-4-5-6-8-9-2

Jalur 6 : 1-2-3-4-5-6-7





# Testing dan Implementasi Sistem

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Teknik Testing

### Test Case untuk contoh di atas :

Test case jalur 1 :

Harga (k) = input valid, dimana  $k < i$  yang ditetapkan di bawah

Harga (i) = -999 dimana  $2 \leq i \leq 100$

Hasil yang diharapkan : rata-rata yang benar berdasarkan nilai k dan total yang tepat.

Catatan : jalur 1 tidak dapat diuji sendiri karena harus diuji sebagai bagian dari pengujian jalur 4,5 dan 6

Test case jalur 2 :

Harga (i) = -999

Hasil yang diharapkan : rata-rata -999, total yang lain pada nilai awal

Test case jalur 3 :

Usahakan untuk memproses 101 nilai atau lebih

100 nilai pertama harus valid

Hasil yang diharapkan : sama seperti test case 1

Test case jalur 4 :

Nilai (i) = input valid dimana  $i < 100$

Nilai (k) < minimum, dimana  $k < i$

Hasil yang diharapkan : rata-rata yang benar berdasarkan nilai-nilai n dan total yang tepat

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Test Case untuk contoh di atas :

Test case jalur 5 :

Nilai (i) = input valid dimana  $i < 100$

Nilai (k) > maksimum, dimana  $k \leq i$

Hasil yang diharapkan : rata-rata yang benar berdasarkan nilai-nilai n dan total yang tepat

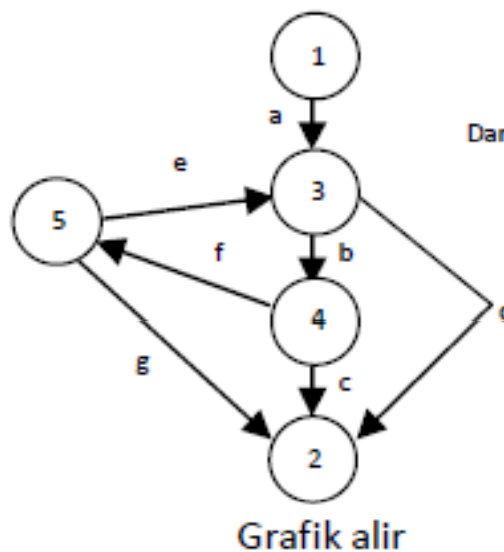
Test case jalur 6 :

Nilai (i) = input valid dimana  $i < 100$

Hasil yang diharapkan : rata-rata yang benar berdasarkan nilai-nilai ni dan total yang tepat.

## Matriks Grafik

- Adalah matriks persegi yang ukurannya (jumlah, baris dan kolom) sama dengan jumlah node pada grafik alir.
- Setiap baris dan kolom terkait dengan node diidentifikasi dan masukan matriks terkait dengan koneksi-koneksi (edge) yang berada di antara node-node



Grafik alir

Dari simpul

	ke simpul				
	1	2	3	4	5
1			a		
2					
3		d		b	
4		c			f
5		g	e		

Matriks grafik

## Matriks Grafik

- Dengan menambahkan sebuah link weight pada masing-masing entri matriks, maka matriks grafis dapat menjadi alat yang sangat kuat untuk mengevaluasi struktur kontrol program selama pengujian.
- Link weight memberikan informasi tambahan mengenai aliran kontrol.
- Dalam bentuk sederhana, link weight adalah 1 (ada hubungan), atau 0 (tidak ada hubungan).

- Contoh :

	1	2	3	4	5
1			a		
2					
3		d		b	
4		c			f
5		g	e		

	ke simpul					
Dari simpul	1	2	3	4	5	Konekal
1			1			1-1=0
2						
3		1		1		2-1=1
4		1			1	2-1=1
5		1	1			2-1=1

3+1=4 ..... Kompleksitas siklomatik

## Pengujian Struktur Kontrol

### – Pengujian Kondisi

- Adalah metode perancangan test case yang menguji kondisi logis yang terkandung dalam modul program.
- Kondisi sederhana adalah variabel Boolean atau ekspresi relasional

$$E_1 < \text{operator-relational} > E_2$$

Dimana E adalah ekspresi aritmetika dan <operator relational> adalah salah satu dari :  $\leq$ ,  $<$ ,  $=$ ,  $>$  dan  $\geq$ .

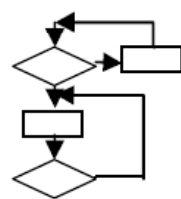
- Kondisi gabungan terdiri atas dua atau lebih kondisi sederhana, operator Boolean (AND, OR, NOT) dan tanda kurung
- Kondisi tidak benar, karena :
  - Kesalahan operator Boolean (hilang/salah/tambahan)
  - Kesalahan variabel Boolean
  - Kesalahan kurung Boolean
  - Kesalahan operator relasional
  - Kesalahan ekspresi aritmatika

## – Pengujian Aliran Data

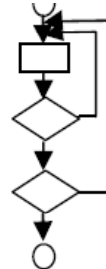
- Adalah pemilihan jalur-jalur uji dari sebuah program menurut lokasi-lokasi dari definisi dan penggunaan variabel dalam program
- Asumsi, setiap pernyataan dalam program diberi sebuah nomor pernyataan yang unik dan setiap fungsi tidak mengubah parameter atau variabel globalnya. Misal pernyataan dengan S sebagai nomor pernyataan :
  - $DEF(S) = \{ X \mid \text{pernyataan } S \text{ mengandung definisi } X \}$
  - $USE(S) = \{ X \mid \text{pernyataan } S \text{ mengandung penggunaan } X \}$Jikalau pernyataan S adalah sebuah pernyataan IF atau loop, himpunan DEF adalah kosong dan himpunan USE didasarkan pada kondisi dari pernyataan S.
- Sebuah rantai penggunaa-definisi (definition use/DU) dari variabel X, mengambil bentuk  $[X, S, S']$ , dimana S dan S' adalah nomor pernyataan, X ada dalam  $DEF(S)$  dan  $USE(S')$ , dan definisi X dalam pernyataan S tinggal di dalam pernyataan S'

## – Pengujian Perulangan

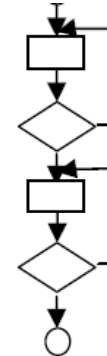
- Adalah teknik pengujian kotak putih / white testing yang memfokuskan secara eksklusif pada validitas konstruksi perulangan.
- Terdapat 4 kelas perulangan : perulangan sederhana, perulangan tersambung, perulangan bersaarang, perulangan tidak terstruktur.



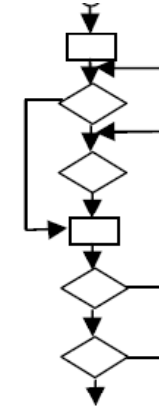
Loop sederhana



Loop tersarang



Loop terangkai



Loop tidak terstruktur

## – Pengujian Perulangan

- **Loop sederhana.** Himpunan berikut harus diaplikasikan pada loop sederhana, diman  $n$  adalah jumlah maksimum yang diijinkan melewati loop tersebut.
  1. Abaikan keseluruhan loop
  2. Hanya satu yang melewati loop
  3. Dua yang melewati loop
  4.  $m$  melewati loop diman  $m < n$
  5.  $n - 1, n, n+1$  melewati loop
- **Loop tersarang.** Bila kita ingin memperluas pendekatan pengujian bagi loop sederhana ke loop tersarang, jumlah pengujian mungkin akan berkembang secara geometris sesuai tingkat penambahan persarangan sehingga sejumlah pengujian menjadi tidak praktis. Beizer mengusulkan suatu pendekatan yang membantu mengurangi jumlah pengujian :
  1. Mulai pada loop yang paling dalam. Atur semua loop ke nilai minimum.
  2. Lakukan pengujian loop sederhana untuk loop yang paling dalam sementara menjaga loop yang paling luar pada nilai parameter iterasi minimumnya (misal pencacah loop). Tambahkan pengujian yang lain untuk nilai out of range atau nilai yang tidak diperbolehkan
  3. Bekerja menuju ke luar, dengan melakukan pengujian untuk loop selanjutnya, tetapi menjaga semua loop bagian luar yang lain pada nilai minimumnya dan loop tersarang lainnya pada harga “tertentu”.
  4. Lanjutkan sampai semua loop telah teruji.



Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## – Pengujian Perulangan

- **Loop terangkai.** Loop terangkai dapat diuji dengan menggunakan pendekatan yang ditentukan untuk loop sederhana bila masing-masing dari loop itu independen terhadap yang lain. Tetapi bila dua loop dirangkai dan pencacah loop untuk loop 1 digunakan sebagai harga awal untuk loop 2, kemudian loop tersebut menjadi tidak independen, maka pendekatan diaplikasikan ke loop tersarang direkomendasi.
- **Loop tidak terstruktur.** Kapan saja memungkinkan kelas loop ini harus didesain lagi untuk mencerminkan penggunaan konsepsi pemrograman terstruktur.

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

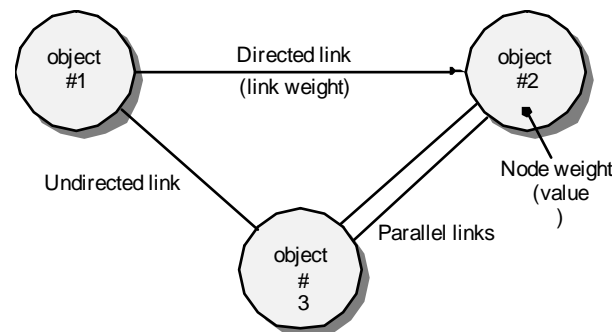
Suplement

## PENGUJIAN KOTAK HITAM/BLACK BOX

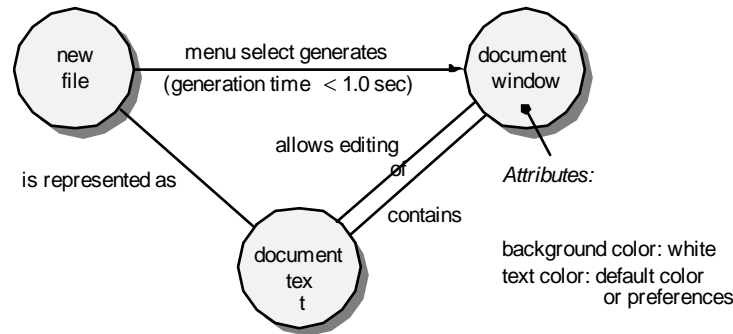
- Disebut juga Pengujian Perilaku
- Berfokus pada persyaratan fungsional perangkat lunak, artinya memungkinkan kita untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program.
- Berupaya menemukan kesalahan untuk kategori :
  1. Fungsi yang salah atau hilang
  2. Kesalahan antar muka
  3. Kesalahan dalam struktur data atau akses basis data eksternal
  4. Kesalahan perilaku atau kinerja
  5. Kealahan inialisasi dan penghentian

## Metode Pengujian Berbasis Grafik

- Bertujuan untuk memahami objek-objek yang dimodelkan dalam perangkat lunak dan penghubung antar objek-objek tersebut
- Langkah :
  - Buat kumpulan node yang mewakili benda
  - Buat tautan/link/edge yang mewakili hubungan antara objek
    - Tautan berarah/directed link
    - Tautan dua arah/simetris
    - Tautan paralel
  - Tentukan bobot node yang menggambarkan sifat dari sebuah node
  - Tentukan bobot tautan yang menggambarkan karakteristik penghubung



– Contoh :



Objek #1 = newFile (pilihan menu)

Objek#2 = documentWindow

Objek#3 = documentText

- Pilihan menu (newFile) menghasilkan jendela dokumen
- Bobot node documentWindow menyediakan daftar atribut yang seharusnya diharapkan ketika jendela dihasilkan
- Bobot tautan menunjukkan jendela harus dibangkitkan dalam waktu kurang dari 1,0 detik
- Tautan tidak berarah membentuk hubungan simetris antara newFile dan documentText

## Partisi Kesetaraan / Equivalence Partitioning

- Membagi ranah masukan program ke dalam kelas-kelas data dari test case yang dapat diturunkan.
- Perancangan test case untuk mempartisi kesetaraan didasarkan pada evaluasi terhadap kelas-kelas kesetaraan untuk kondisi masukan
- Jika sekumpulan objek dapat dihubungkan dengan hubungan yang simetris, transitif dan refleksif maka terdapat kelas kesetaraan
- Sebuah kelas kesetaraan merepresentasikan keadaan valid atau tidak valid dari kondisi-kondisi masukan
- Kondisi masukan dapat berupa nilai numerik yang spesifik, rentang nilai, seperangkat nilai-nilai yang terkait atau kondisi Boolean

## Partisi Kesetaraan / Equivalence Partitioning

Pedoman kelas-kelas kesetaraan :

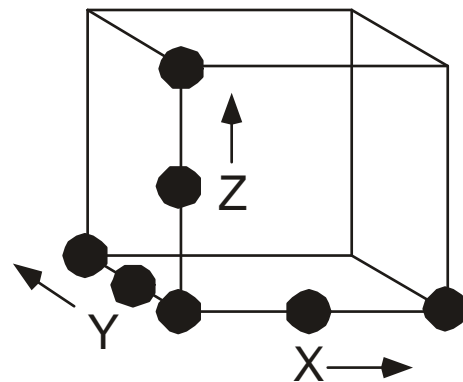
- Jika kondisi masukan menspesifikasikan kisaran, satu kelas kesetaraan yang valid dan dua kelas kesetaraan yang tidak valid didefinisikan
- Jika suatu kondisi masukan membutuhkan nilai tertentu, satu kelas kesetaraan yang valid dan dua kelas kesetaraan yang tidak valid didefinisikan
- Jika kondisi masukan menspesifikasikan anggota dari himpunan, satu kelas kesetaraan yang valid dan satu yang kelas kesetaraan tidak valid didefinisikan
- Jika kondisi masukan adalah Boolean, satu kelas kesetaraan yang valid dan satu kelas kesetaraan yang tidak valid ditentukan

## Analisis Nilai Batas

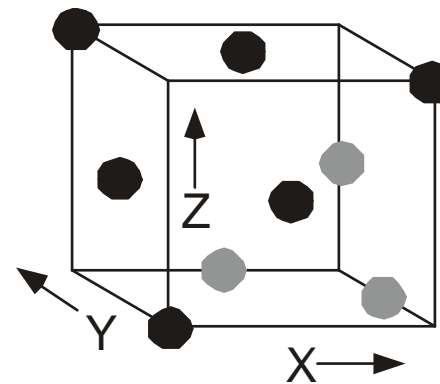
- Mengarah ke seleksi test case yang menguji nilai-nilai batas
- Pelengkap partisi kesetaraan
- Pedoman analisis nilai batas
  1. Jika kondisi masukan menspesifikasikan kisaran yang dibatasi oleh nilai a dan b, test case harus dirancang dengan nilai a dan b dan hanya di atas dan di bawah a dan b
  2. Jika kondisimasukan menspesifikasikan sejumlah nilai , test case harus dikembangkan untuk menguji jumlah-jumlah minimum dan maksimum
  3. Terapkan pedoman 1 dean 2 untuk kondisi keluaran
  4. Jika struktur data program internal memiliki batas-batas yang telah ditentukan, pastika untuk merancang sebuah test case untuk menguji struktur data pada batasnya.

## Pengujian Larik Ortogonal

- Diterapkan untuk masalah-masalah dimana domain masukan relatif kecil tapi terlalu besar untuk mengakomodasi pengujian lengkap
- Bermanfaat untuk menemukan kesalahan sebuah wilayah-sebuah kategori kesalahan yang terkait dengan logika yang salah dalam komponen PL



One input item at a time



L9 orthogonal array



Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Pengujian Berbasis Model

- Teknik pengujian menggunakan informasi yang termuat dalam model persyaratan sebagai dasar bagi pembuatan test case
- Menggunakan diagram state dalam UML
- Langkahnya :
  1. Menganalisis model perilaku yang sudah ada untuk perangkat lunak atau membuatnya jika belum ada
  2. Jelajahi model perilaku dan tentukan masukan yang akan memaksa PL untuk melakukan transisi dari keadaannya ke keadaan lainnya.
  3. Lakukan peninjauan terhadap model perilaku dan perhatikan kelaur yang diharapkan ketika PL membuat transisi dari satu keadaan ke keadaan lainnya
  4. Jalankan test case
  5. Bandingkan hasil aktual dan hasil yang diharapkan dan ambil tindakan korektif saat diperlukan

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## PENGUJIAN UNTUK LINGKUNGAN, ARSITEKTUR DAN APLIKASI KHUSUS

### Pengujian GUI / Graphical User Interface

- Dapat menggunakan finite state modelling graph untuk memperoleh serangkaian pengujian yang menunjukkan data yang spesifikasi dan objek program yang relevan dengan GUI (= pengujian berbasis model)

Teknik Testing

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Pengujian Arsitektur Klien Server

- Lingkungan klien server bersifat terdistribusi, potensi masalah : platform perangkat keras yang berbeda, kompleksitas komunikasi jaringan, kebutuhan layanan beberapa klien dari basis data terpusat (atau terdistribusi), dan persyaratan koordinasi yang dikenakan pada semua server.
- Tiga tingkatan pengujian :
  1. Aplikasi-aplikasi klien individu diuji dalam mode terputus; operasi server dan jaringan mendasarinya tidak dipertimbangkan
  2. PL klien dan aplikasi-aplikasi server yang terkait diuji bersama-sama namun operasi jaringan tidak secara eksplisit dieksekusi
  3. Arsitektur klien server yang lengkap termasuk operasi dan kinerja jaringan diuji.
- Pendekatan Pengujian :
  - **Pengujian fungsi aplikasi.**
  - **Pengujian server.** Koordianasi dan fungsi manajemen data server diuji, demikian pula kinerja server (waktu respon dan trhougput)
  - **Pengujian Basis data.** Keakuratan dan integritas data yang disimpan di server diuji
  - **Pengujian transaksi.** Dilakukan serangkaian pengujian utnuk memastikan bahwa masing-masing kelas transaksi diproses sesuai dengan kebutuhan
  - **Pengujian jaringan komunikasi.** Pengujian untuk membutkikan bahwa komunikasi antara node=node jaraingan benar terjadi dan pesan lewat, transaksi dan lalu lintas jairngan yang terkait terjadi tanpa kesalahan

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

## Pengujian Sistem Waktu Nyata / Real Time

- Sifat waktu nyata : bergantung waktu dan tak sinkron
- Perancangan test case, selain mempertimbangkan test case konvensional, juga penanganan kejadian (event handling) : pengolahan interupsi, pewaktuan data, paralelisme tugas/proses yang menangani data
- Langkah :
  1. Pengujian tugas. Menguji masing-masing tugas secara independen
  2. Pengujian perilaku. Menggunakan model sistem yang diciptakan dengan alat bantu otomatis untuk mensimulasikan sistem waktu nyata dan memeriksa perilakunya sebagai konsekuensi dari peristiwa eksternal
  3. Pengujian antartugas. Setelah kesalahan diisolasi, pengujian beralih ke kesalahan yang terkait dengan waktu. Tugas berkomunikasi melalui antrian pesan atau penyimpanan data untuk diuji dan menemukan kesalahan dalam penetapan ukuran area penyimpanan data
  4. Pengujian sistem. PL dan PK diintegrasikan dan serangkaian penuh pengujian sistem dilakukan dalam upaya untuk menemukan kesalahan antarmuka PL-PK.

Kontrak  
Perkuliahan

Review  
Rekayasa  
Perangkat Lunak

Manajemen  
Kualitas

Strategi &  
Teknik Testing

Implementasi  
Sistem

Suplement

Teknik Testing

## Pertanyaan Dan Diskusi

.....