

CONTOH PENGUJIAN  
BLACK BOX DAN WHITE  
BOX

# Contoh Black Box Testing dengan Equivalence Partitioning

Pemeliharaan data untuk aplikasi bank yang sudah diotomatisasikan. Pemakai dapat memutar nomor telepon bank dengan menggunakan mikro komputer yang terhubung dengan password yang telah ditentukan dan diikuti dengan perintah-perintah. Data yang diterima adalah :

- Kode area : kosong atau 3 digit
- Prefix : 3 digit atau tidak diawali 0 atau 1
- Suffix : 4 digit
- Password : 6 digit alfanumerik
- Perintah : check, deposit, dll

Selanjutnya kondisi input digabungkan dengan

# Contoh Black Box Testing dengan Equivalence Partitioning

- Kode area : kondisi input, Boolean -kode area mungkin ada atau tidak kondisi input, range -nilai ditentukan antara 200 dan 999
  - Prefix : kondisi input range > 200 atau tidak diawali 0 atau 1
  - Suffix : kondisi input nilai 4 digit
- Password : kondisi input boolean – password mungkin diperlukan atau tidak kondisi input nilai dengan 6 karakter string
  - Perintah : kondisi input set berisi perintah-perintah yang telah didefinisikan

**Table pengujian black-box pada validasi login admin aplikasi sistem pakar**

No.	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Mengosongkan semua isian data login, lalu langsung mengklik tombol 'Masuk'.	Nama: - Kata sandi: -	Sistem akan menolak akses login dan menampilkan pesan "Mohon isi dulu nama admin dan kata sandi"	Sesuai harapan	Valid
2.	Hanya mengisi data nama admin dan mengosongkan data kata sandi, lalu langsung mengklik tombol 'Masuk'.	Nama: admin Kata sandi: -	Sistem akan menolak akses login dan menampilkan pesan "Mohon isi dulu salah satu data yang masih kosong"	Sesuai harapan	Valid
3.	Hanya mengisi data kata sandi dan mengosongkan data nama admin, lalu langsung mengklik tombol 'Masuk'.	Nama: - Kata sandi: 123	Sistem akan menolak akses login dan menampilkan pesan "Mohon isi dulu salah satu data yang masih kosong"	Sesuai harapan	Valid
4.	Menginputkan dengan kondisi salah satu data benar dan satu lagi salah, lalu langsung mengklik tombol 'Masuk'.	Nama: mimin (salah). Kata sandi: 123 (benar).	Sistem akan menolak akses login dan menampilkan pesan "Nama pengguna atau kata sandi salah! Silakan ulangi dengan data yang benar"	Sesuai harapan	Valid
5.	Menginputkan sintaks SQL Injection pada kedua data login, lalu mengklik tombol 'Masuk'.	Nama: ' OR '1'='1 Kata sandi: ' OR '1'='1	Sistem akan menolak akses login dan menampilkan pesan "Nama pengguna atau kata sandi salah! Silakan ulangi dengan data yang benar"	Sesuai harapan	Valid

# Tabel pengujian black-box pada validasi login admin aplikasi sistem pakar

No.	Kondisi Input	Data Input	Kondisi Output	Ket
6.	Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'.	Nama: admin Kata sandi: 123	Sistem menerima akses login dan kemudian langsung menampilkan form pakar/admin.	Sesuai harapan Valid

## Keterangan:

- = kosong (tidak diisi).

Contoh di atas hanyalah pada salah satu *use case* atau satu fungsi, yaitu fungsi login admin. Anda bisa kembangkan lebih lanjut sesuai dengan kasus pengguna atau fungsi-fungsi lain pada aplikasi yang hendak Anda uji.

# Contoh Testing White Box 1:

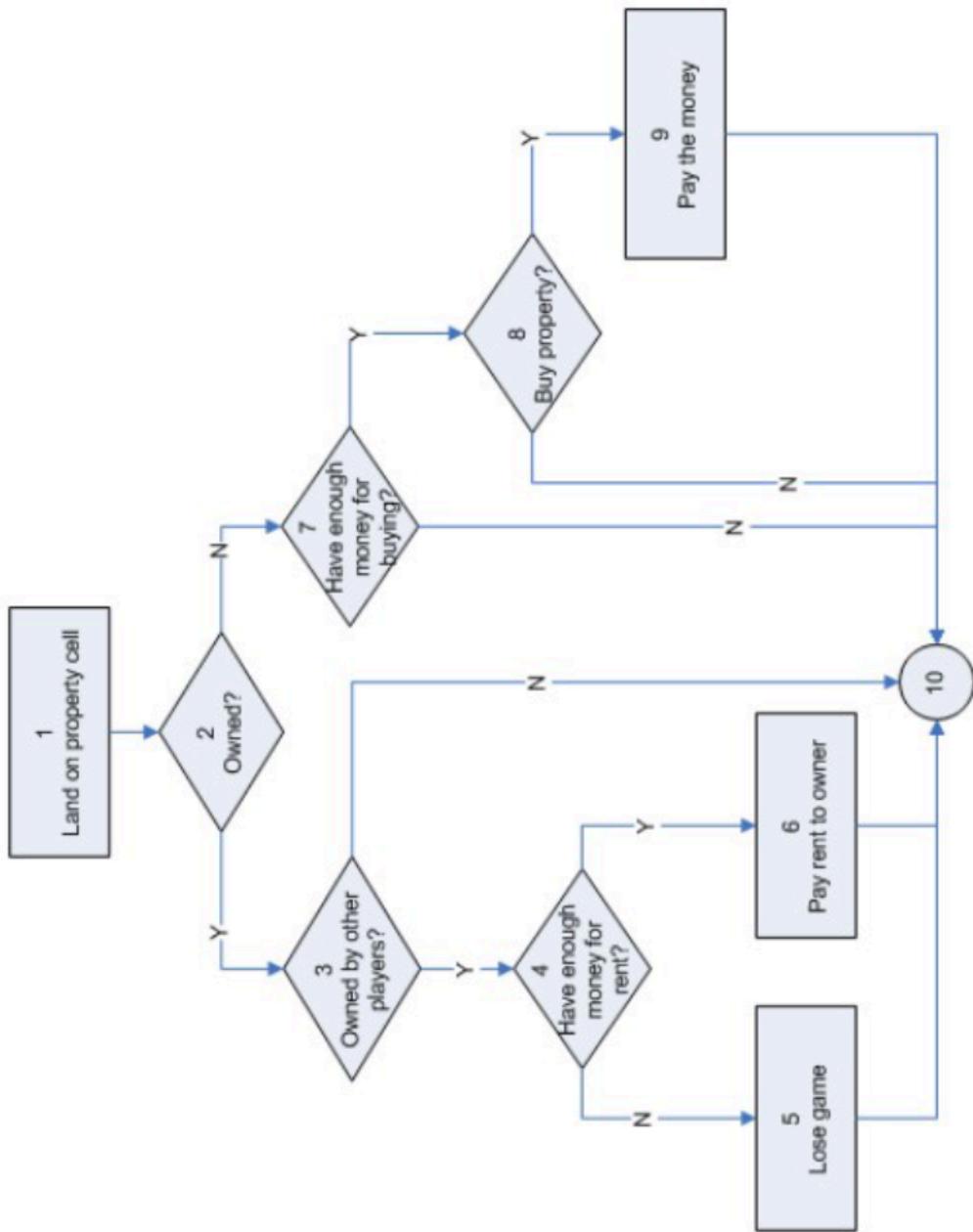


Figure 1: Flowgraph of purchasing property

# Contoh Testing White Box 1

- Menggunakan grafik aliran ini, kita dapat menghitung jumlah jalur independen melalui kode. Kami melakukan ini dengan menggunakan metrik disebut nomor cyclomatic (McCabe, 1976), yang didasarkan pada teori grafik.
- Cara termudah untuk menghitung jumlah siklomatik adalah dengan menghitung jumlah *conditional / predikat* (diamond) dan tambahkan 1. Dalam contoh di atas, ada lima *conditional*. Oleh karena itu, jumlah *cyclomatic* kami adalah 6, dan kami memiliki enam jalur independen melalui kode. Jadi kita sekarang dapat menghitungnya:

# Contoh Testing White Box 1

- 1. 1-2-3-4-5-10 (properti yang dimiliki oleh orang lain, tidak mempunyai uang untuk sewa)
- 2. 1-2-3-4-6-10 (properti yang dimiliki oleh orang lain, membayar sewa)
- 3. 1-2-3-10 (properti yang dimiliki oleh pemain)
- 4. 1-2-7-10 (properti yang tersedia, tidak memiliki cukup uang)
- 5. 1-2-7-8-10 (properti yang tersedia, punya uang, tidak ingin membelinya)
- 6. 1-2-7-8-9-10 (properti yang tersedia, punya uang, dan membelinya)

# Contoh Testing White Box 1

- Kami ingin menulis kasus pengujian untuk memastikan bahwa setiap jalur yang akan diuji setidaknya sekali. Seperti dikatakan di atas, jumlah siklomatik adalah batas bawah pada jumlah kasus uji yang akan kita tulis. Uji kasus yang ditentukan dengan cara ini adalah yang kami gunakan dalam pengujian basis patch.

# Contoh Pengujian White Box

2:

**Step 1:** Contoh prosedur di bawah ini menunjukkan bagaimana laporan algoritma dipetakan ke node grafik, nomor di sebelah kiri.

public double calculate(int amount)

```
{  
-1-    double rushCharge = 0;  
-1-    if (nextday.equals("yes"))  
{  
-2-        rushCharge = 14.50;  
    }  
-3-    double tax = amount * .0725;  
-3-    if (amount >= 1000)  
{  
-4-        shipcharge = amount * .06 + rushCharge;  
    }  
-5-    else if (amount >= 200)  
{  
}
```

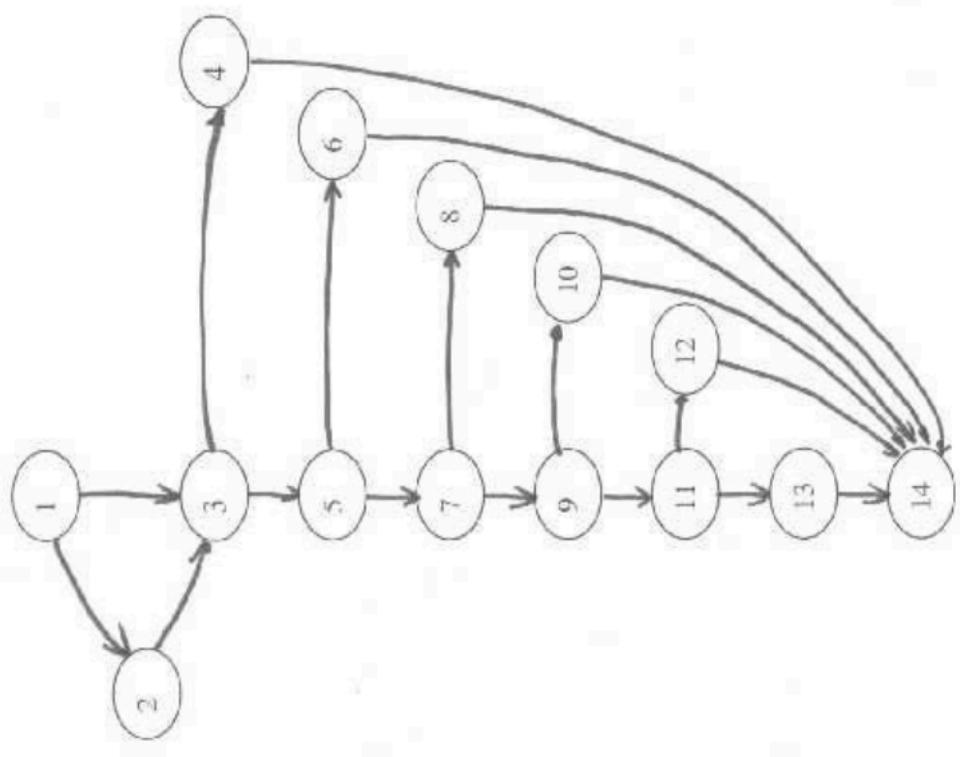
# Contoh Pengujian White Box

2:

```
{  
-6-    shipcharge = amount * .08 + rushCharge;  
}  
-7- else if (amount >= 100)  
{  
-8-    shipcharge = 13.25 + rushCharge;  
}  
-9- else if (amount >= 50)  
{  
-10-    shipcharge = 9.95 + rushCharge;  
}  
-11- else if (amount >= 25)  
{  
-12-    shipcharge = 7.25 + rushCharge;  
}
```

```
else
{
    -13-    shipcharge = 5.25 +
rushCharge;
}
-14- total = amount + tax +
shipcharge;
-14- return total;
} //end calculate
```

Dibawah ini adalah flowchart dari contoh program diatas :



**Step 2:** Menentukan kompleksitas cyclomatic dari grafik aliran.

$$\begin{aligned}V(G) &= E - N + 2 \\&= 19 - 14 + 2 \\&= 7\end{aligned}$$

**Keterangan:**

E : Jumlah Busur atau Link

N : Jumlah Simpul

Ini menjelaskan bahwa batas atas pada ukuran basis set. Artinya, memberikan jumlah jalur independen yang perlu kita cari.

- **Step 3:** Menentukan dasar jalur independen

Path 1: 1 - 2 - 3 - 5 - 7 - 9 - 11 - 13 -  
14

Path 2: 1 - 3 - 4 - 14

Path 3: 1 - 3 - 5 - 6 - 14

Path 4: 1 - 3 - 5 - 7 - 8 - 14

Path 5: 1 - 3 - 5 - 7 - 9 - 10 - 14

Path 6: 1 - 3 - 5 - 7 - 9 - 11 - 12 - 14

Path 7: 1 - 3 - 5 - 7 - 9 - 11 - 13 - 14

- **Step 4:** Menyiapkan test cases bahwa pelaksanaan kekuatan setiap jalur di set dasar.

<u>path</u>	<u>nextday</u>	<u>amount</u>	<u>expected result</u>
1	yes	10	30.48
2	no	1500	????.??
3	no	300	345.75
4	no	150	174.125
5	no	75	90.3875
6	no	30	39.425
7	no	10	15.975

Penyataan pengulangan statement di tengah-tengah blok diperlukan meskipun ada gambaran sampai akhir, jika itu adalah simbol terminal tambahan.

Kembali dari ekspresi boolean diperlakukan jika ada penyataan (*statement*)