

Black box testing



ISQA

Definisi

- Black Box testing merupakan strategi testing dimana hanya memperhatikan/memfokuskan kepada faktor fungsionalitas dan spesifikasi perangkat lunak.
- Berbeda dengan white box, black box testing tidak membutuhkan pengetahuan mengenai, alur internal (internal path), struktur atau implementasi dari software under test (SUT).

Kategori Kesalahan yang Diuji oleh Black Box Testing

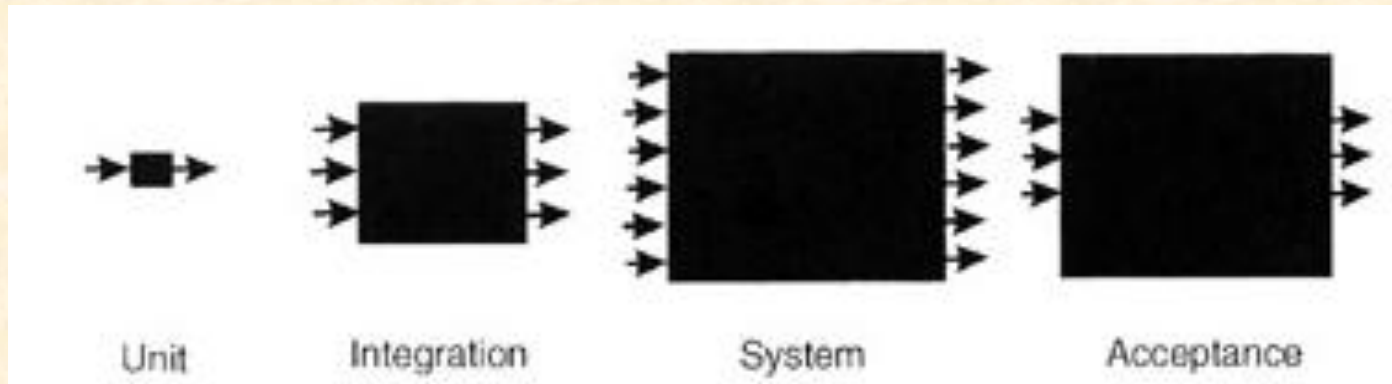
1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan performa.
5. Kesalahan inisialisasi dan terminasi

Proses Black Box Testing

- Menganalisis kebutuhan dan spesifikasi dari perangkat lunak.
- Pemilihan jenis input yang memungkinkan menghasilkan output benar serta jenis input yang memungkinkan output salah pada perangkat lunak yang sedang diuji.
- Menentukan output untuk suatu jenis input.
- Pengujian dilakukan dengan input-input yang telah benar-benar diseleksi.
- Melakukan pengujian.
- Perbandingan output yang dihasilkan dengan output yang diharapkan.
- Menentukan fungsionalitas yang seharusnya ada pada perangkat lunak yang sedang diuji.

Ruang Lingkup Black Box

- *Black Box testing* dapat dilakukan pada setiap level pembangunan sistem. Mulai dari *unit, integration, system, dan acceptance*.



Keunggulan dan kelemahan Black Box

- Black box testing dapat menguji keseluruhan fungsionalitas perangkat lunak.
- Black box testing dapat memilih *subset test* yang secara efektif dan efisien dapat menemukan cacat. Dengan cara ini *black box testing* dapat membantu memaksimalkan *testing investment*.
- Ketika *tester* melakukan *black box testing*, *tester* tidak akan pernah yakin apakah perangkat lunak yang diuji telah benar-benar lolos pengujian.

Equivalence Class Testing

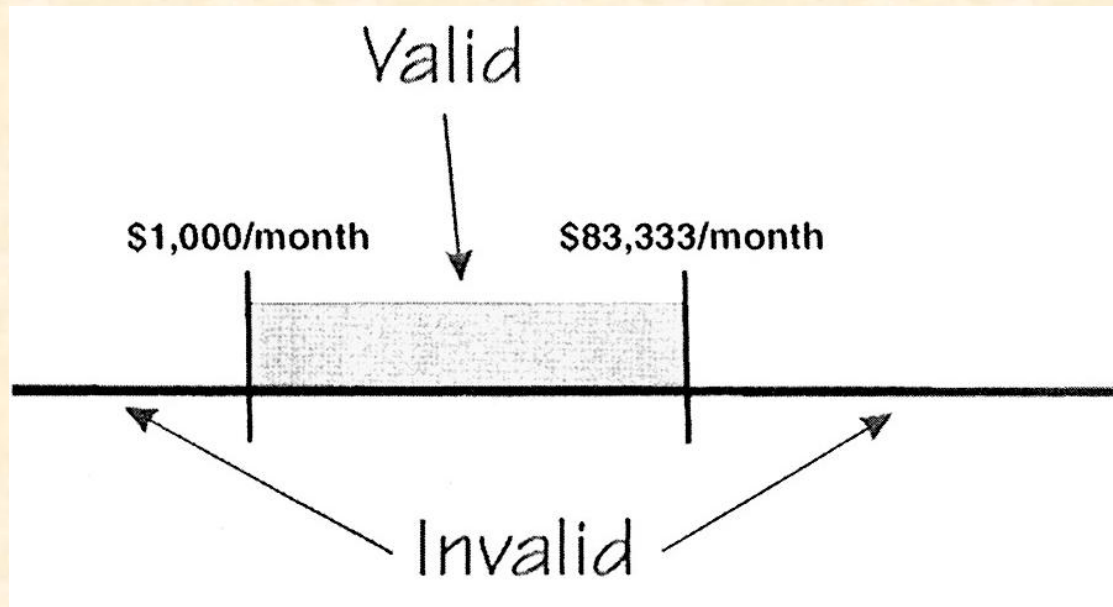
- Merupakan teknik yang digunakan untuk mengurangi jumlah test case yang ada pada saat pengujian. Kebanyakan tester menggunakan teknik yang simpel ini meskipun secara formal tester tersebut tidak mengetahui mengenai metode desain formal dalam pengujian perangkat lunak.
- Kasus uji yang didesain untuk *Equivalence class testing* berdasarkan pada evaluasi dari ekuivalensi jenis/class untuk kondisi input. *Class-class yang ekuivalen* merepresentasikan sekumpulan keadaan valid dan invalid untuk kondisi input.
 - Biasanya kondisi input dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi boolean.

Langkah-langkah *Equivalence Class Testing*

- Identifikasi kelas-kelas yang ekuivalen (*equivalence class*).
- Buat *test case* untuk tiap-tiap *equivalence class*.
- Jika memungkinkan buat test case tambahan yang acak yang memungkinkan ditemukannya cacat pada perangkat lunak. (yg sangat menyimpang)

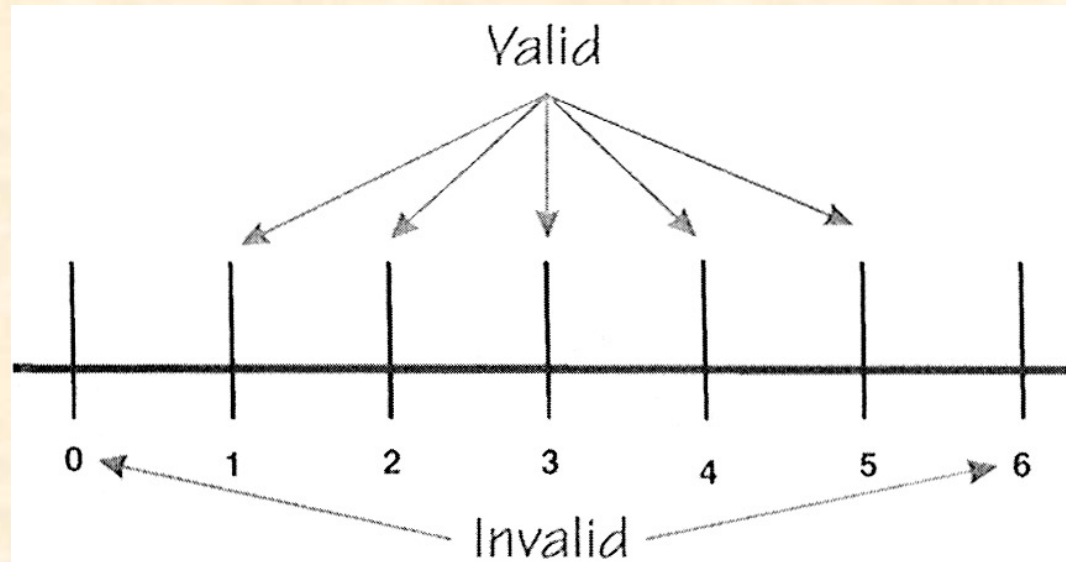
Continuous equivalence classes

Contoh nilai untuk pendapatan/ *salary* yang disyaratkan untuk melakukan pembelian rumah secara kredit:



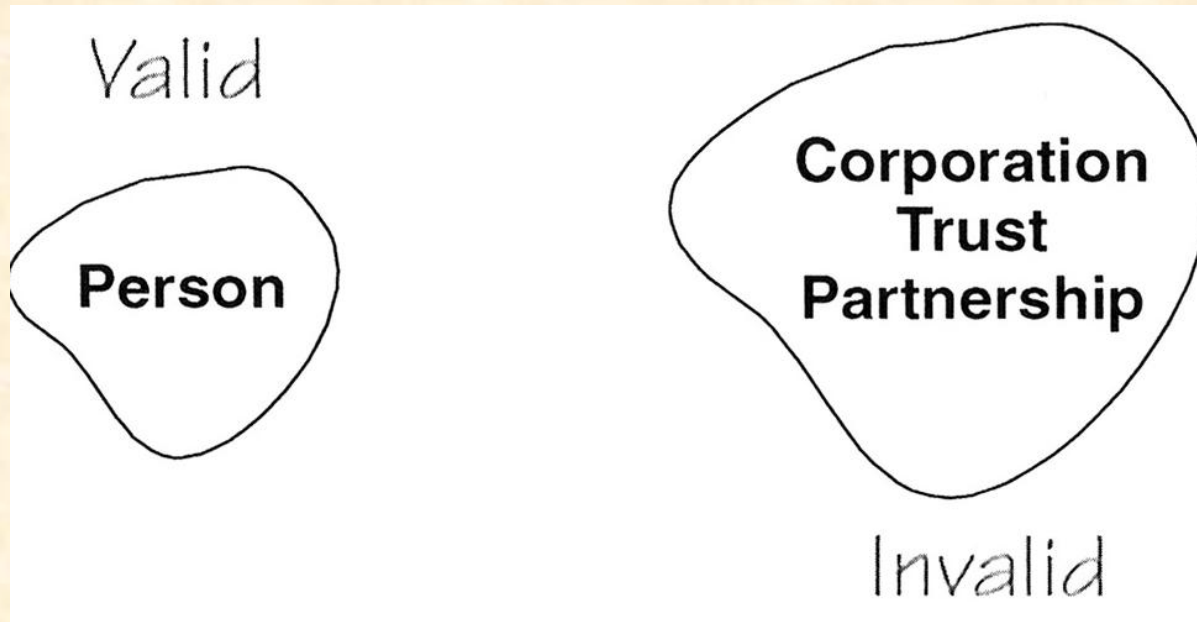
Discrete equivalence classes

Contoh nilai untuk jumlah kemilikan rumah yang disyaratkan untuk melakukan pembelian rumah secara kredit:



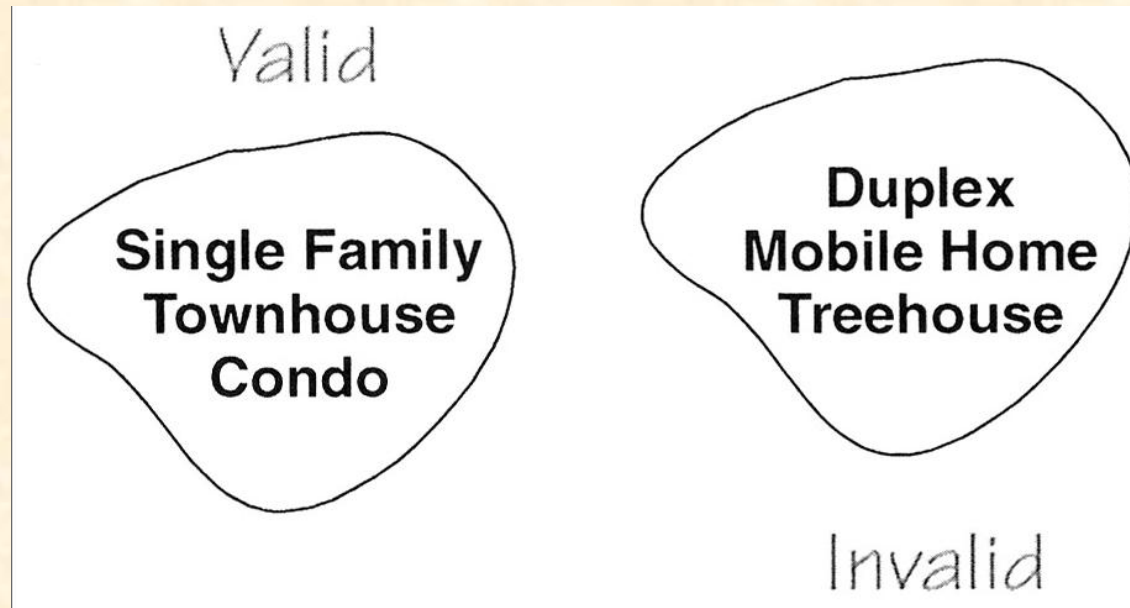
Single selection equivalence classes

Contoh nilai untuk kategori pengajuan yang disyaratkan untuk melakukan pembelian rumah secara kredit:



Multiple selection equivalence class

Contoh nilai untuk jenis rumah yang disyaratkan untuk melakukan pembelian rumah secara kredit:



Boundary Value Testing

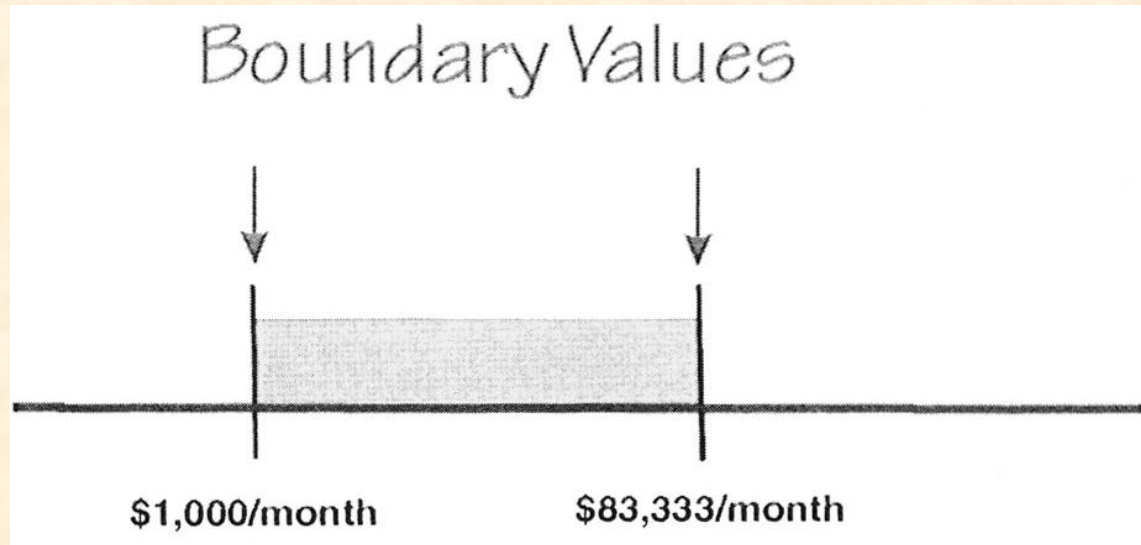
- *Boundary value testing (BVT)* fokus kepada suatu batasan nilai dimana kemungkinan terdapat cacat yang tersembunyi. (tepat di batas)
- BVT mengarahkan pada pemilihan kasus uji yang melatih nilai-nilai batas. BVT merupakan desain teknik kasus uji yang melengkapi *Equivalence class testing*. Dari pada memfokuskan hanya pada kondisi input, BVA juga menghasilkan kasus uji dari domain output.

Langkah-langkah Boundary Value Testing

- Identifikasi kelas-kelas yang ekuivalen (*equivalence class*).
- Identifikasi batasan untuk tiap *equivalence class*.
- Buat *test case* untuk tiap batasan suatu nilai dengan memilih titik pada batasan, satu titik pada nilai bawah batasan dan satu titik pada nilai atas batasan.

Boundary values for a continuous range of inputs

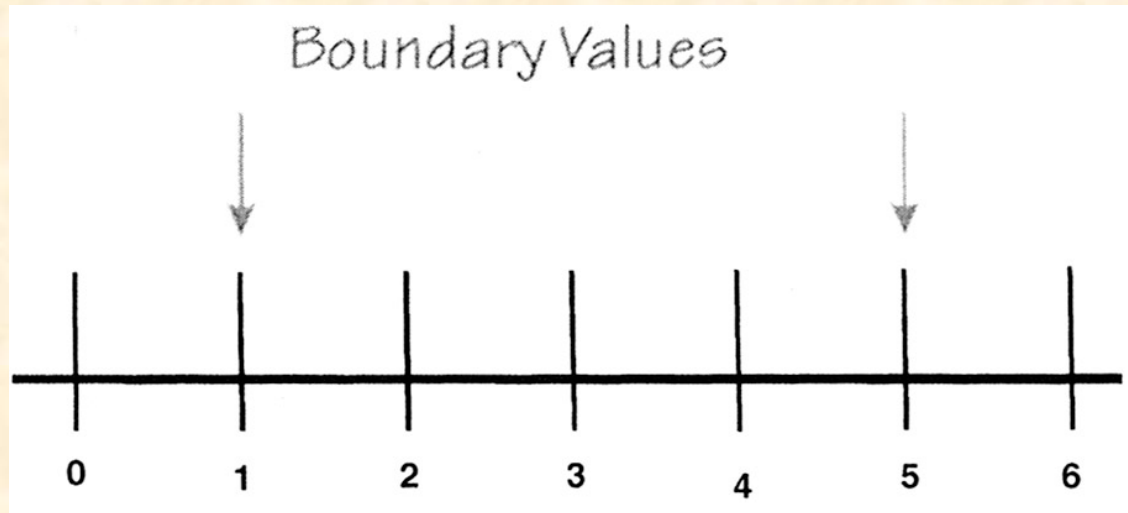
Contoh untuk nilai pendapatan/ salary:



Tes data input untuk batas bawah adalah {\$999, \$1,000, \$1,001} dan untuk batas atas {\$83,332, \$83,333, \$83,334}.

Boundary values for a discrete range of inputs.

Contoh nilai untuk jumlah tempat tinggal (dwellings) yang dimiliki oleh seseorang:



Tes data input untuk batas bawah adalah $\{0, 1, 2\}$ dan untuk batas atas $\{4, 5, 6\}$.

Contoh Kombinasi Pengujian

- Sangat penting untuk menginputkan nilai kombinasi secara bersamaan, misalkan:

Monthly Income	Number of Dwellings	Result	Description
\$1,000	1	Valid	Min income, min dwellings
\$83,333	1	Valid	Max income, min dwellings
\$1,000	5	Valid	Min income, max dwellings
\$83,333	5	Valid	Max income, max dwellings
\$1,000	0	Invalid	Min income, below min dwellings
\$1,000	6	Invalid	Min income, above max dwellings
\$83,333	0	Invalid	Max income, below min dwellings
\$83,333	6	Invalid	Max income, above max dwellings
\$999	1	Invalid	Below min income, min dwellings
\$83,334	1	Invalid	Above max income, min dwellings
\$999	5	Invalid	Below min income, max dwellings
\$83,334	5	Invalid	Above max income, max dwellings

Holodeck

- Holodeck is a unique test tool that uses fault injection to simulate real-world application and system errors for Windows applications and services - allowing testers to work in a controlled, repeatable environment to analyze and debug error-handling code.
- Holodeck is the first commercial fault-simulation tool and was developed by leading researchers in the application quality field. It is used by organizations like Microsoft, Adobe, EMC and McAfee.

Holodeck

The screenshot displays the Holodeck Enterprise Edition application window. The title bar reads "contoh - Holodeck Enterprise Edition". The menu bar includes "File", "Session", "Application", "Log", "Tools", "View", and "Help". The toolbar contains various icons for file operations and help. The main workspace is divided into four columns of fault categories, each with a list of radio button options:

- Disk Faults:** No Disk Fault, Too Many Open Files, File Not Found, File Locked, File Is Read Only, File In Use, File Cannot Be Created, File Already Exists, Drive Cannot Seek Disk, Disk Full, Data Error, Cannot Remove Directory, Access Denied To File (selected).
- Memory Faults:** No Memory Fault, Segment Locked, Page File Too Small, Invalid Address, Invalid Access, Insufficient Memory (selected).
- Network Faults:** No Network Fault (selected), Wrong Version of Winsock, Winsock Task Limit Reached, No Ports Available, Network Not Installed, Network Down, Network Disconnected.
- Registry Faults:** No Registry Fault, Value Not Found, Value Cannot Be Written, Value Cannot Be Read, Registry Is Corrupt, Query Buffer Too Small, No More Query Items, No Log Space, Key Not Found, Key Marked For Deletion, Key Is Corrupt, Key Cannot Be Opened, IO Operation Failed, Access Denied (selected).

On the right side, there is a "Project" pane showing a tree view of fault categories: Faults, Disk, Access, Memory, Insuffi, Network, Registry, Access, Process, Access, Custom, Resource Fau, and Limits. Below it is a "Dynamic Help" pane with the following text:

Fault Pane Help
Faults are preset error conditions that will help you test your application easily. There are six categories including one to create your own faults to test your application.

At the bottom, there is a "Limits" pane showing a progress bar for "Memory Space". The "Used" value is 65668 and the "Limit" is 1694800. The units are set to "KB".

The taskbar at the bottom shows icons for "Limits", "Resources", "File Corruption Details", and "Network Message Details".

Contoh kasus-1

- Program mencatat pembayaran sejumlah barang dari tabel barang. Jika data valid akan menghitung nilai bayar jika data tdk valid akan muncul pesan kesalahan. Data yang diinput adalah kode barang dan jml barang. Isi tabel barang : kd_barang,nama brg, satuan, harga
- Buatlah kelas ekivalensi test case utk menguji program tsb.

Kelas Ekivalensi

- Asumsi kode barang 5 digit:
 - A=atk
 - B=barang lain
(jml digit kode barang = A/B xxxx)
- Data valid
 - kode brg hrs look-up tabel
 - jml brg integer >0

Kelas Ekivalensi

- Data invalid
 - kode brg diinput baru
 - kode brg blank
 - jumlah brg ≤ 0
 - jml brg pecahan / non numerik

Berapa kali dilakukan test case?
(valid = 3, invalid = 4)

Test Case/Kasus Uji -> pake tabel pengujian

No	Data uji	Input	Hasil tes diharapkan	Output	Kesiimpulan
1	Kd brg=lookup tabel Jml>0	Kd brg = A1111 Jml=2	Perhitungan nilai bayar	Perhitungan nilai bayar	Ok/sukses
2	Kd brg=kosong, jml>0	Kd brg= ` ` Jml=2	Pesan kesalahan	Pesan kesalahan	Ok
3	Kd -brg=isi baru Jml >0	Kd brg=b9999 Jml=3	Pesan kesalahan	Hitung nilai barang	Invalid / tdk OK

Kasus-2

- Program mencatat nilai uts mahasiswa dari suatu kuliah. Jika data valid akan menyimpan nilai UTS ke tabel nilai, jika data tdk valid akan muncul pesan kesalahan.
 - Data yang diinput adalah mata kuliah (kd-kuliah), NPM dan nilai UTS (integer).
 - Isi tabel mahasiswa : NPM, nama mhs.
 - Isi tabel nilai: npm, kd-kuliah, nUTS
- Buatlah kelas ekivalensi dan test case utk menguji program tsb.

Kelas Ekuivalensi kasus 2

- Asumsi

- Data valid

- Kode kuliah harus look-up tabel
- NPM harus look-up tabel
- Nilai UTS=integer, $0 \leq \text{NUTS} \leq 100$

Kelas Ekuivalensi kasus 2

- Data invalid
 - Kode kuliah blank
 - NPM blank
 - Kode kuliah diinput baru
 - NPM diinput baru
 - Nilai UTS <0 atau Nilai UTS >100
 - NUTS=real / non-numerik / blank

Tabel pengujian kasus 2

Asumsi :

NO	Data Uji	Input	Hasil yg diharapkan	Output testing	Hasil uji/kesimpulan
2	Kd kul & NPM lookup, 0<NUTS>10 0	Kd kul & NPM lookup, NUTS = 75	Data Tersimpan	Data Tersimpan	OK
3	Kd kul & NPM diisi baru, 0<NUTS>10 0	Kd kul & NPM diisi baru, NUTS = 75	Pesan kesalahan	Data Tersimpan	Tdk OK
4	...				
5	...				

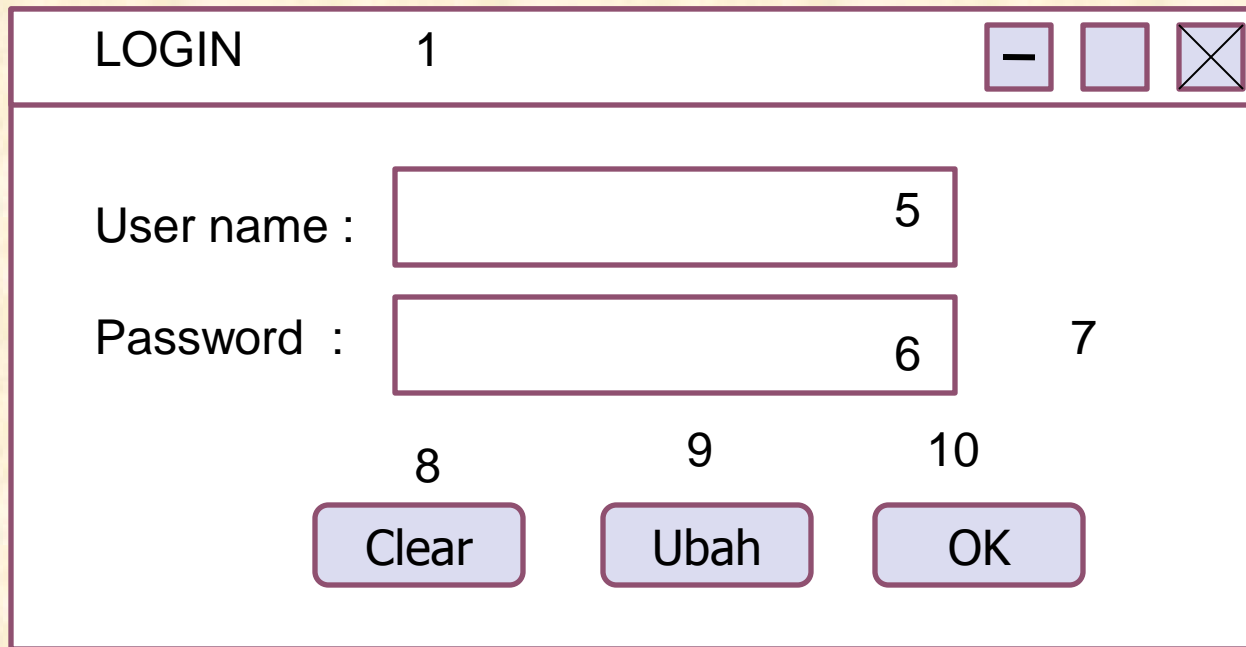
Latihan (buat tabel pengujiannya)

1. Program untuk menginput 2 bilangan bulat kemudian mencari nilai maksimum dr kedua bil tsb.
2. Program untuk menampilkan keterangan terbilang dari bilangan /uang (bulat positif) yang diinput. Nilai terbesar adalah 5 milyar

Contoh :

- Jumlah uang : 1650
- Terbilang : seribu enam ratus lima puluh

GUI test → black box



Tabel pengujian terdiri dari :

No	Input	Hasil yang diharapkan	Output tes	Hasil Uji
1	Klik no.2	Kotak dialog menjadi minimum		