



**UNIVERSITAS
BINA INSANI**

TESTING DAN IMPLEMENTASI SISTEM

OBJECT-ORIENTED TESTING

Ahmad Chusyairi, M.Kom



binainsani.ac.id



Bina Insani University



[binainsaniuniversity](https://www.instagram.com/binainsaniuniversity)



[@BinaInsaniOK](https://twitter.com/BinaInsaniOK)

Jl. Raya Siliwangi No. 6 Rawa Parjang - Bekasi



Capaian Pembelajaran Pertemuan

- Mahasiswa mengetahui dan memahami *object-oriented testing*

Kemampuan Akhir Capaian Pembelajaran

- Mahasiswa mampu menjelaskan dan memahami *object-oriented testing*

Bahan Kajian

- *Object-Oriented Testing*

Penilaian

- Mahasiswa mampu menjelaskan *object-oriented testing*



Object-Oriented Testing

- 🦄 Proses ujicoba sistem yang berorientasi objek dimulai dengan meninjau ulang analisis dan model desain berorientasi obyeknya (*object-oriented analysis and design models*).
- 🦄 Ketika sebuah program telah dituliskan, *object-oriented testing* (OOT) dimulai dengan menguji *in the small* dengan class testing (*class operations* dan *collaborations*).
- 🦄 Ketika *class-class* tersebut diintegrasikan menjadi sebuah subsistem, maka masalah kolaborasi class akan diketahui.
- 🦄 Terakhir, use cases dari model OOA digunakan untuk menemukan kesalahan validasi software.





Object-Oriented Testing (2)








- OOT hampir mirip dengan ujicoba software konvensional dalam hal kasus uji yang akan dibangun untuk melatih *class-class* yang ada dan kolaborasi antar *class*-nya juga perilakunya.
- OOT berbeda dari ujicoba software konvensional dalam hal penekanan terhadap konsistensi dan kelengkapan penaksiran dari model OOA dan OOD yang telah dibangun.
- OOT cenderung lebih fokus kepada masalah integrasi daripada *unit testing*.



OOT Activities




- 🦿 Meninjau ulang model OOA dan OOD.
- 🦿 Ujicoba *class* setelah penulisan program sumber.
- 🦿 Ujicoba integrasi dalam subsistem.
- 🦿 Ujicoba integrasi subsistem yang telah ditambahkan ke dalam sistem.
- 🦿 Ujicoba validasi berdasarkan OOA use-cases

Integration Testing dalam Konteks OO

-  Difokuskan pada kelompok-kelompok kelas yang berkolaborasi atau berkomunikasi dalam beberapa cara.
-  Integrasi operasi satu persatu ke dalam kelas sering tidak optimal.
-  Ujicoba berbasis *thread* (uji semua kelas yang dibutuhkan untuk merespon ke satu masukan atau *event* sistem).
-  Pengujian berbasis kegunaan (dimulai dengan uji independen oleh kelas pertama dan kelas-kelas yang tergantung yang menggunakannya).
-  Pengujian *cluster* (kerjasama kelompok kelas yang diuji untuk interaksi kesalahan).
-  Pengujian regresi adalah penting karena setiap *thread*, *cluster*, atau subsistem yang ditambahkan pada sistem.
-  Tingkat integrasi yang lebih sedikit berbeda dalam sistem berorientasi objek.



Validation Testing dalam Konteks OO

-  Berfokus pada tindakan pengguna yang terlihat dan pengguna dapat mengenali output dari sistem.
-  Tes validasi didasarkan pada skenario *use-case*, model perilaku objek, dan diagram alur *event* dibuat dalam model OOA.
-  Pengujian Black-box konvensional dapat digunakan untuk mendorong tes validasi.

Test Case Design untuk Software OO

- 🐼 Setiap kasus uji harus dapat diidentifikasi secara unik dan eksplisit dihubungkan dengan *class* yang akan diujikan.
- 🐼 Tetapkan kegunaan dari setiap ujicoba.
- 🐼 Tuliskan langkah-langkah ujicoba dan untuk setiap ujicoba disertakan:
 - 🔥 Tuliskan tahapan ujicoba untuk setiap objek yang disertakan dalam ujicoba.
 - 🔥 Tuliskan pesan-pesan dan operasi yang dijalankan sebagai konsekuensi dari ujicoba ini.
 - 🔥 Tuliskan eksepsi yang muncul ketika suatu objek di ujicoba.
 - 🔥 Tuliskan kondisi eksternal yang memerlukan perubahan untuk ujicoba tersebut.
 - 🔥 Informasi tambahan lainnya untuk mengimplementasikan ujicoba tersebut.



Ujicoba Struktur Permukaan dan Struktur Dalam

- 🦄 Ujicoba struktur permukaan (*Testing Surface Structure*), yaitu melatih struktur yang tampak oleh pengguna akhir, sering melibatkan pengamatan dan mewancarai pengguna karena mereka memanipulasi objek sistem.
- 🦄 Ujicoba struktur dalam (*Testing Deep Structure*), yaitu melatih struktur program internal, seperti ketergantungan, perilaku, dan mekanisme komunikasi yang ada sebagai bagian dari sistem dan desain objek.



Inter-Class Test Case Design

- ❖ Desain kasus uji menjadi lebih rumit seperti halnya integrasi dari dimulainya sistem OO untuk menguji kolaborasi antar *class*.
- ❖ Ujicoba *class* yang beragam, seperti:
 - 🔥 Untuk setiap *class client* dengan daftar operator class untuk men-*generate* urutan uji coba *random* yang mengirimkan pesan ke *server class* yang lain.
 - 🔥 Untuk setiap pesan, tentukan tingkatan operator.
- ❖ Uji yang dihasilkan dari model perilaku:
 - 🔥 Gunakan *State Transition Diagram* (STD) sebagai model yang merepresentasikan perilaku dinamis dari suatu class.
 - 🔥 Kasus uji harus mencakup seluruh tahapan STD.







Inter-Class Test Case Design (2)

- 🔥 *Breadth first traversal* dari *state model* dapat digunakan (uji satu transisi dalam satu waktu dan hanya membuat kegunaan dari transisi yang diujikan sebelumnya ketika mengujikan transisi yang baru).
- 🔥 Kasus uji juga dapat dihasilkan untuk memastikan bahwa seluruh perilaku untuk *class* telah diujikan dengan benar.



Testing Methods Applicable at Inter-Class Level

Cluster Testing

-  Menitikberatkan pada pengintegrasian dan menguji kelompok objek yang bekerjasama.
-  Mengidentifikasi kelompok dengan menggunakan pengetahuan tentang pengoperasian objek dan memiliki sistem yang diterapkan oleh kelompok ini.

Pendekatan *Cluster Testing*


Use-Case atau Ujicoba Skenario

-  Ujicoba berdasarkan pada interaksi user dengan sistem.
-  Mempunyai keuntungan bahwa fitur ujicoba sistem seperti yang dialami pengguna.




Testing Methods Applicable at Inter-Class Level (2)

Thread Testing

-  Menguji respon sistem terhadap *event* sebagai pemrosesan *thread* melalui sistem.

Object Interaction Testing

-  Menguji urutan interaksi objek yang berhenti ketika operasi objek tidak memanggil layanan dari objek lainnya.

Use Case / Scenario-based Testing

- 🦄 Berdasarkan pada:
 - 🔥 *Use cases*
 - 🔥 *Corresponding sequence diagrams*
- 🦄 Identifikasi skenario dari use-case dan menambahkan hal ini dengan diagram interaksi yang menampilkan objek-objek yang termasuk dalam skenario:
 - 🔥 Konsentrasi pada kebutuhan (fungsional):
 - 🦄 Setiap *use case*.
 - 🦄 Setiap perluasan kombinasi perpanjangan penuh (<>).
 - 🦄 Setiap perluasan kombinasi penggunaan penuh (<>).
 - 🦄 Ujicoba normal seperti perilaku.
- 🦄 Sebuah skenario adalah sebuah alur atau *path* melalui diagram urutan.
- 🦄 Banyak skenario yang berbeda dapat dihubungkan dengan sebuah diagram urutan.

Use Case / Scenario-based Testing (2)

- 🦁 Menggunakan tugas pengguna yang dideskripsikan dalam *use-cases* dan membangun kasus uji dari tugas-tugas tersebut dan variasinya.
- 🦁 Menemukan kesalahan yang muncul ketika pengguna berinteraksi dengan software OO.
- 🦁 Konsentrasi pada apa yang dilakukan oleh fungsinya, bukan apa yang dilakukan oleh produknya.
- 🦁 Dapat memperoleh nilai kembali yang lebih besar atas usaha dan waktu lebih yang dikeluarkan dalam meninjau ulang *use-cases* semenjak mereka dibuat, daripada membuang waktu untuk uji coba *use-case*.



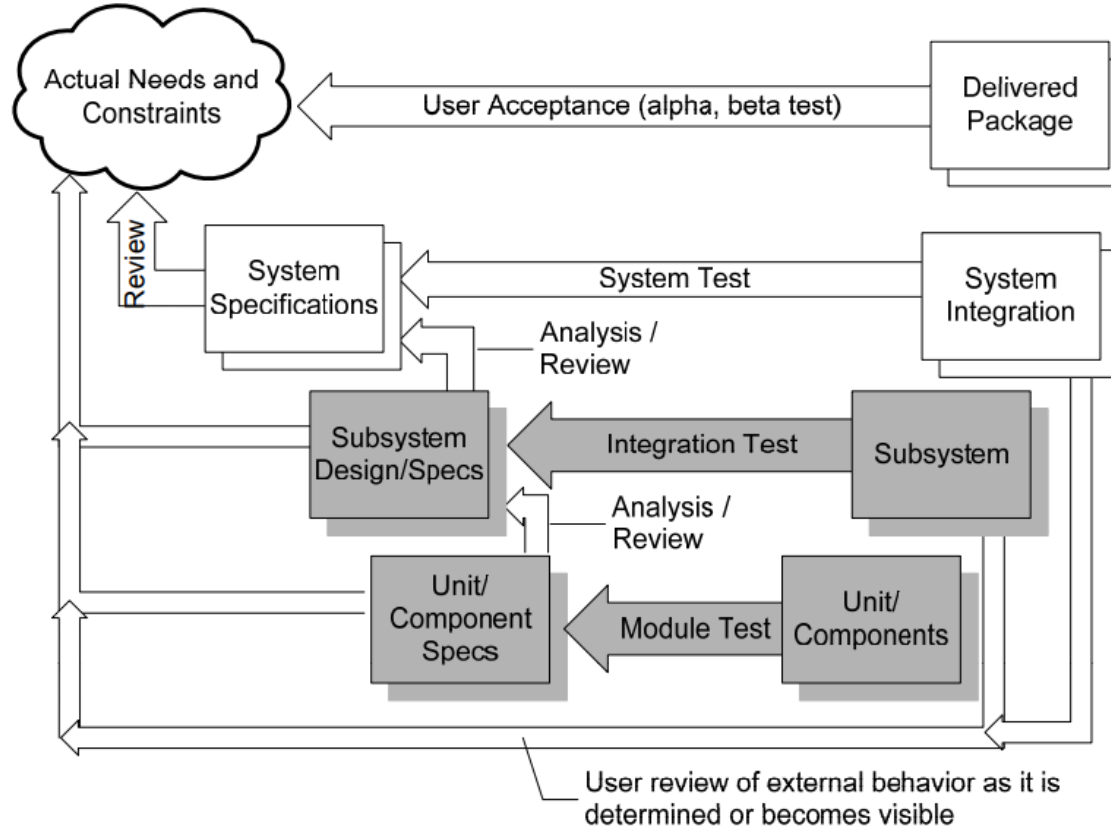
OO Test Design Issues

- Metode *White-Box Testing* dapat diaplikasikan pada ujicoba kegunaan program untuk mengimplementasi operasi class tetapi tidak untuk lainnya.
- Metode *Black-Box Testing* tepat untuk ujicoba sistem OO.
- Object-Oriented Programming* menyebabkan kebutuhan tambahan untuk ujicoba, berupa:
 - Class* dapat mengandung operasi yang diturunkan dari *super classes*.
 - Subclass* dapat mengandung operasi yang didefinisikan ulang daripada diturunkan.
 - Seluruh *class* dihasilkan dari ujicoba *class* dasar sebelumnya memerlukan ujicoba lanjutan.



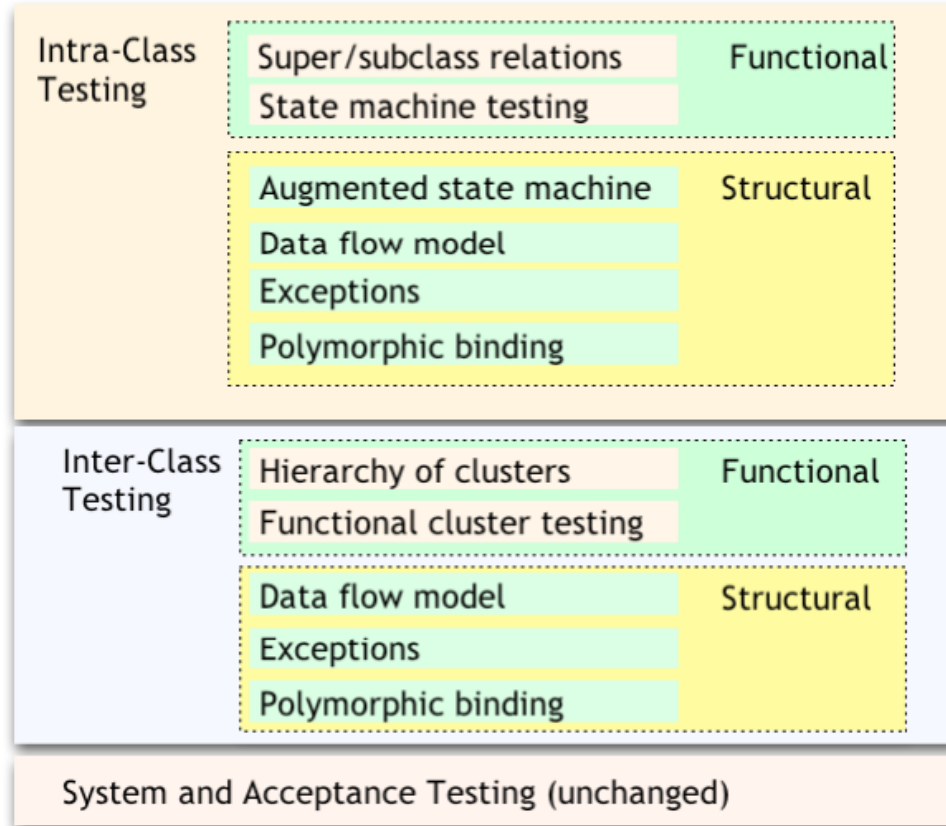


Quality Activities and OO SW



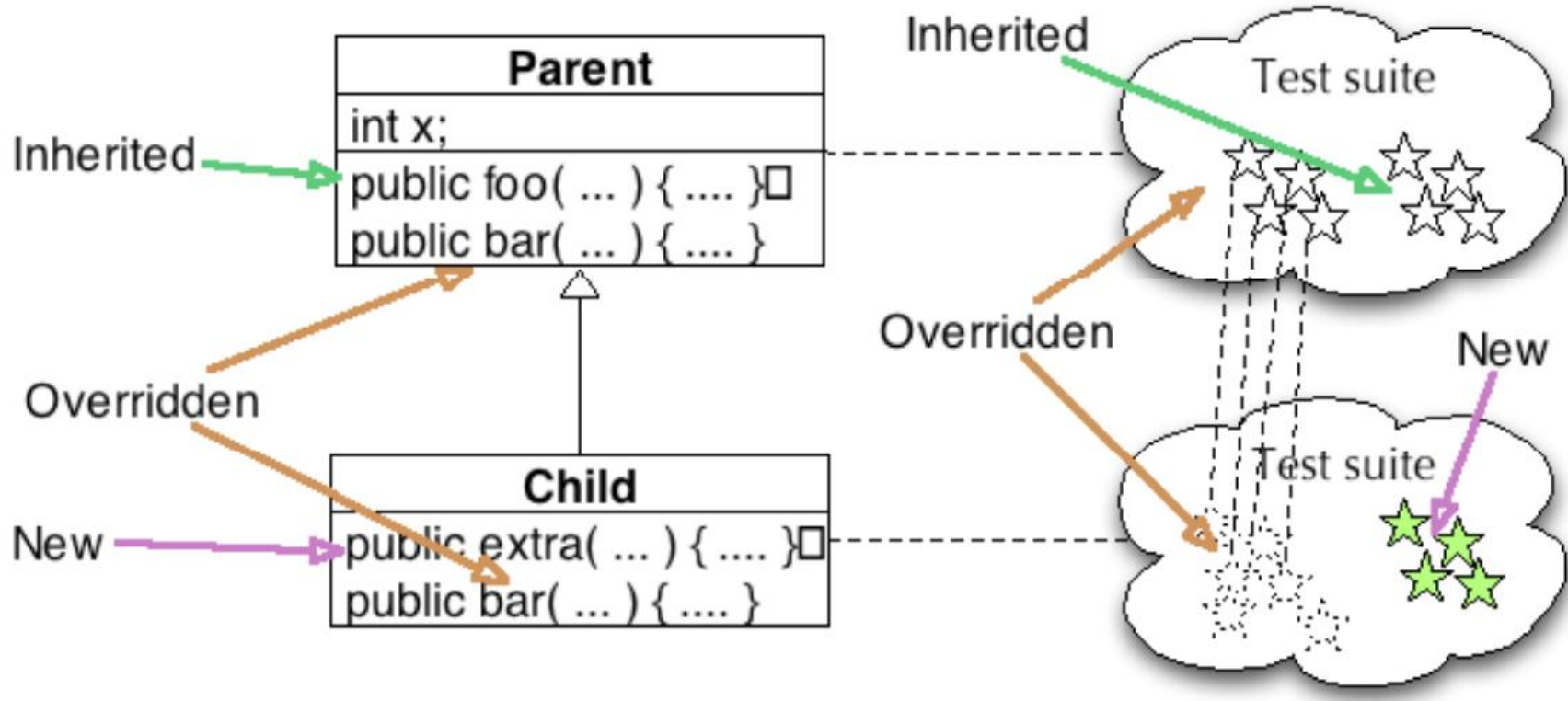


Orthogonal Approach: Stages





Testing History-Summary





Exception Handling

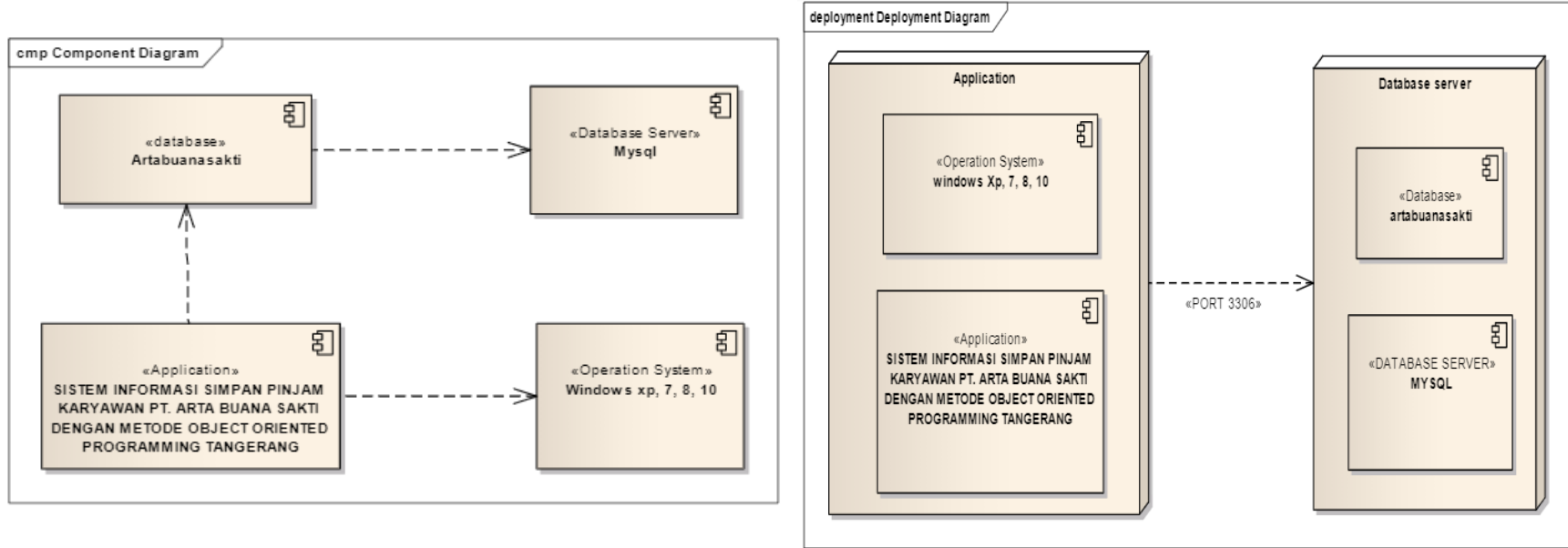
```
void addCustomer(Customer theCust) {
    customers.add(theCust);
}

public static Account
newAccount(...)
throws InvalidRegionException
{
    Account thisAccount = null;
    String regionAbbrev = Regions.regionOfCountry(
        mailAddress.getCountry());

    if (regionAbbrev == Regions.US) {
        thisAccount = new USAccount();
    } else if (regionAbbrev == Regions.UK) {
        ...
    } else if (regionAbbrev == Regions.Invalid) {
        throw new InvalidRegionException(mailAddress.getCountry());
    }
    ...
}
```

Pengecualian membuat aliran kontrol implisit dan dapat ditangani oleh penanganan yang berbeda

Rancang Bangun Sistem Informasi Simpan Pinjam Karyawan Menggunakan Metode Object Oriented Programming (Studi Kasus: PT. Arta Buana Sakti Tangerang)

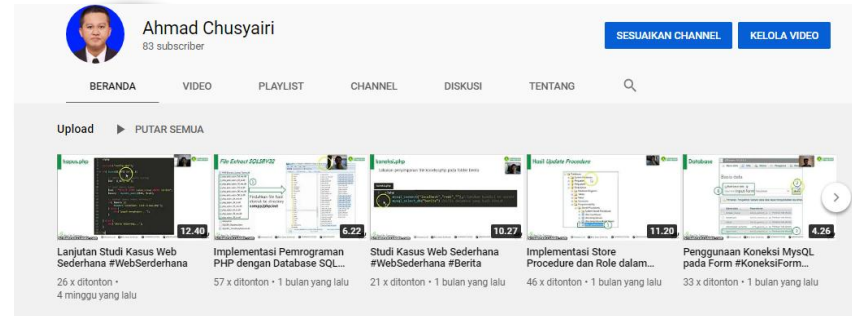


- Cayunda F. N. Pengujian Object Oriented. Retrieved from URL: [https://www.academia.edu/24769754/Pengujian Object Oriented](https://www.academia.edu/24769754/Pengujian_Object_Oriented) (diakses tanggal 30-04-2021).
- Fridayanthie E. W. And Charter J. Rancang Bangun Sistem Informasi Simpan Pinjam Karyawan Menggunakan Metode Object Oriented Programming (Studi Kasus: Pt. Arta Buana Sakti Tangerang), Jurnal Techno Nusa Mandiri, 13(2), pp.149-156, 2016.
- Pezze M and Young M. Testing Object Oriented Software Chapter 15. Retrieved from URL: <https://ix.cs.uoregon.edu/~michal/book/slides/ppt/PezzeYoung-Ch15-OOTesting.ppt> (diakses tanggal 30-04-2021).

Terima Kasih

Bigger
Better
Higher

 **ahmad chusyairi**



Ahmad Chusyairi
83 subscriber

[SESUAIKAN CHANNEL](#) [KELOLA VIDEO](#)

BERANDA VIDEO PLAYLIST CHANNEL DISKUSI TENTANG

Upload ▶ PUTAR SEMUA

Thumbnail	Video Title	Views	Upload Date
	Lanjutan Studi Kasus Web Sederhana #WebSederhana	26 x ditonton • 4 minggu yang lalu	12:40
	Implementasi Pemrograman PHP dengan Database SQL...	57 x ditonton • 1 bulan yang lalu	6:22
	Studi Kasus Web Sederhana #Berita	21 x ditonton • 1 bulan yang lalu	10:27
	Implementasi Store Procedure dan Role dalam...	46 x ditonton • 1 bulan yang lalu	11:20
	Penggunaan Koneksi MySQL pada Form #KoneksiForm...	33 x ditonton • 1 bulan yang lalu	4:26