

# Pengembangan Aplikasi Web

## Pertemuan Ke-11 & 12 (Keamanan Aplikasi *Web*)

**Noor Ifada**

Email: [noor.ifada@trunojoyo.ac.id](mailto:noor.ifada@trunojoyo.ac.id)

Scopus: [56590032100](#)

Google Scholar: [Noor Ifada](#)

ResearchGate: [Noor-Ifada](#)

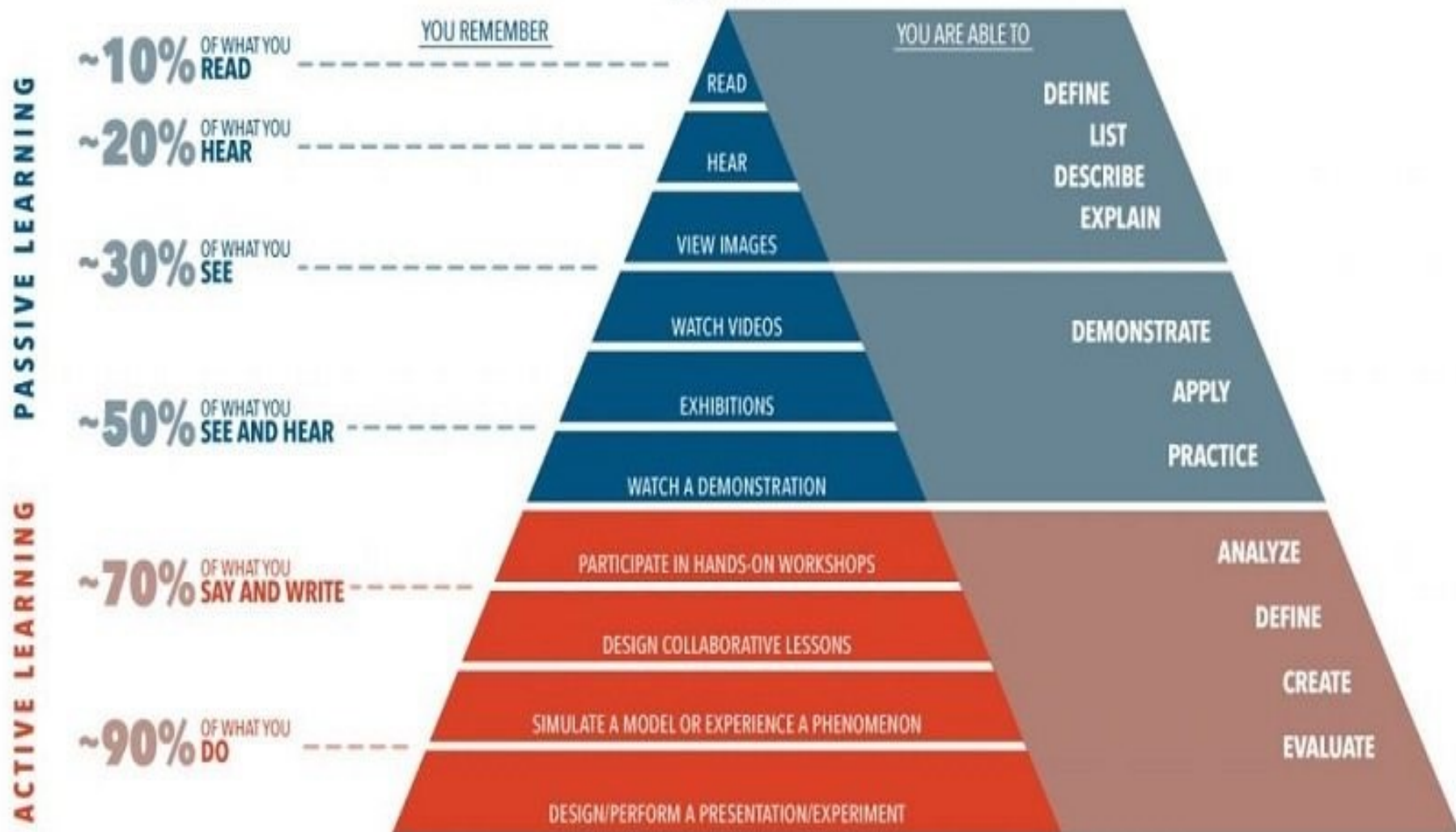
Repository: [Trunojoyoan](#)

Semester Gasal 2023/2024

S1 Teknik Informatika – Universitas Trunojoyo Madura (UTM)

# CONE OF EXPERIENCE

EDGAR DALE



<https://elearningindustry.com/>

# SOFT SKILLS FOR STUDENTS

## 6 SKILLS TO HELP YOU SUCCEED



**1 Problem Solving**

Do you see problems as interesting opportunities? Problem solvers anticipate potential stumbling blocks and act to prevent them or mitigate their effects.

**TIP:** You don't always need to start by trying to solve the problem. First, aim to understand the root of the problem.



**2 Creativity**

Are you able to think outside of the box even when you're not asked to? Everyone has creative abilities. Creative thinking skills help you to look at problems and situations from a fresh perspective.



**3 Communication**

Are you an effective communicator and active listener? Communication skills will help you understand others and be better understood in return. This can help make each interaction you have in both your work and social life a more positive experience.




**4 Time Management**

Are you an organized person who meets deadlines? Good time management skills enable you to do more in less time, helping you reduce stress and maintain a better work-life balance.

**TIP:** Prioritize your tasks and delegate a time limit to how long you spend on each one.



**5 Stress Management**

Can you identify and tackle your stress triggers? Being able to manage stress allows you to break the hold it can have on your life. This will make you happier, healthier, and more productive in return.

**TIP:** Set limits and learn to say no to requests that would create excessive stress in your life.



**6 Teamwork**

Are you a team player who collaborates well with others? Teamwork requires leadership and goal orientation. Whether in school or a job, the better you work within a team, the better your work will be!

**TIP:** Always clarify roles, responsibilities and accountability with your team members.



**END**



You won't always be a priority to others,  
and that's why you have to be a priority to yourself.


Learn to respect yourself, take care of yourself,  
and become your own support system.

Your needs matter. Start meeting them. Don't wait  
for others to choose you. Choose yourself.

*Jazz Zo Marcellus*

# Tabu Bagi Mahasiswa Teknik Informatika

- ❖ Tidak dapat beradaptasi dengan (perkembangan) teknologi
- ❖ Tidak dapat membaca atau memahami instruksi
- ❖ Tidak dapat membuat program



**LEARNING IS NEVER  
DONE WITHOUT  
ERRORS AND DEFEAT**

VLADIMIR LENIN

PICTUREQUOTES.com



PICTUREQUOTES

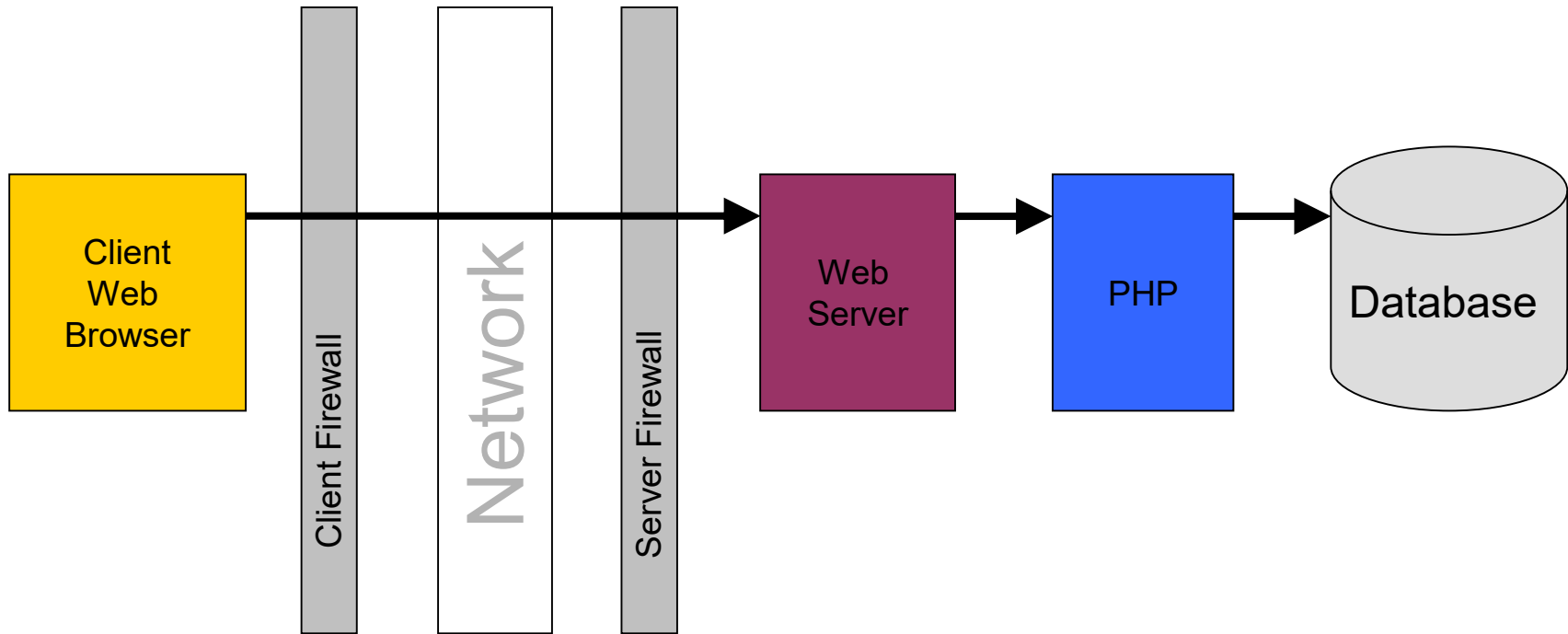
# Sub Pokok Bahasan

- Prinsip Keamanan
- Keamanan *End-to-End*
- Keamanan Basisdata
- Pengamanan *Password*
- Serangan Keamanan Aplikasi *Web*
- Implementasi keamanan yang lemah
- *Session*

# Prinsip Keamanan

- **Confidentiality**
  - Mencegah kebocoran data (data tidak dapat diakses oleh pihak yang tidak berwenang)
- **Integrity**
  - Melindungi konsistensi data
- **Availability**
  - Memastikan bahwa data selalu tersedia ketika dibutuhkan
- **Authenticity**
  - Memvalidasi identitas *user* (pihak yang ingin mengakses data)
- **Non-repudiation**
  - Mencegah penolakan transaksi

# Keamanan *End-to-End*





# Keamanan Basisdata

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Replication Variables Charsets More

User accounts overview User groups

## User accounts overview

⚠ A user account allowing any user from localhost to connect is present. This will prevent other users from connecting if the host part of their account allows a connection from any (%) host. ⓘ

	User name	Host name	Password	Global privileges ⓘ	User group	Grant	Action
<input type="checkbox"/>	Any	%	No	USAGE		No	Edit privileges  Export
<input type="checkbox"/>	Any	localhost	No	USAGE		No	Edit privileges  Export
<input type="checkbox"/>	pma	localhost	No	USAGE		No	Edit privileges  Export
<input type="checkbox"/>	root	127.0.0.1	No	ALL PRIVILEGES		Yes	Edit privileges  Export
<input type="checkbox"/>	root	::1	No	ALL PRIVILEGES		Yes	Edit privileges  Export
<input type="checkbox"/>	root	localhost	No	ALL PRIVILEGES		Yes	Edit privileges  Export

⬆ ☐ Check all With selected: Export

```
1 <?php
2 $dbc = new PDO('mysql:host=localhost;dbname=customerdb','root','');
3
4 // Use the connection ...
5
6 ?>
```

# Pengamanan *Password*

- Tidak menyimpan data *password* dalam bentuk “*clear text*” di dalam basisdata
- Gunakan fungsi/function *hashing* untuk enkripsi data: **md5**, **sha2**, ...

Stored password: **Today123**

SHA2

fa82bc6820fcc9098dff6ea2ec5b829e14944500bab3f4be19d029d41cb8031

```
SELECT * FROM admin
WHERE username = :username
and password = SHA2(:password, 0)
```

Don't Match!

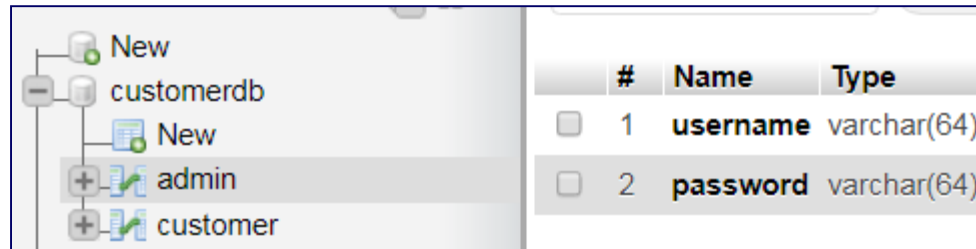
Entered password: **today123**

SHA2

685cb7bec82cc19f15e9a1246a3ad813912a36905f248d26925bd8680a65d5a4

# Pengamanan *Password* [2]

- Buat tabel untuk menyimpan data *user* dan *password*. Contoh: tabel **admin**



The screenshot shows a database management interface. On the left, a tree view displays the database structure: 'New' (database), 'customerdb' (database), 'New' (table), 'admin' (table), and 'customer' (table). The 'admin' table is selected. On the right, the table structure is displayed in a table format:

#	Name	Type
<input type="checkbox"/> 1	<b>username</b>	varchar(64)
<input type="checkbox"/> 2	<b>password</b>	varchar(64)

- Tambahkan data ke tabel **admin**

```
Run SQL query/queries on table customerdb.admin: ?
```

```
1 INSERT INTO admin (username, password) VALUES('Amira', SHA2('Secret!', 0));
```

username	password
Amira	5de772715ff750859d6efa965201bb4ccd059ed04740dd867d...

## Challenge #1

Buatlah tabel “admin” yang memiliki kolom “username” dan “password”

Tambahkan sebuah data baru ke dalam tabel “admin” dimana data kolom “password” dienkripsi dengan menggunakan fungsi **SHA2()**

# Serangan Keamanan Aplikasi Web

- SQL Injection
- Script Injection
- XML Attack

# SQL Injection

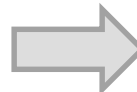
- Jangan menggunakan pernyataan SQL yang menggabungkan masukan *user* secara langsung → gunakan PDO *Prepared Statements*: **prepare()**, **bindValue()**, **execute()**

```
Run SQL query/queries on table customerdb.customer: ?  
1 SELECT *  
2 FROM customer  
3 WHERE customerID=1;
```



customerID	firstname	address	balance
1	Almira	Jl. Mawar No. 123, Surabaya	4500000

```
Run SQL query/queries on table customerdb.customer: ?  
1 SELECT *  
2 FROM customer  
3 WHERE customerID=1 or true;
```



customerID	firstname	address	balance
1	Almira	Jl. Mawar No. 123, Surabaya	4500000
2	Baharudin	Jl. Mengkudu No. 456, Surabaya	1200000
3	Citra	Jl. Cendana No. 17, Surabaya	0
4	Derry	Jl. Delima No. 50, Surabaya	0
5	Erlina	Jl. Erlangga No. 44, Surabaya	0

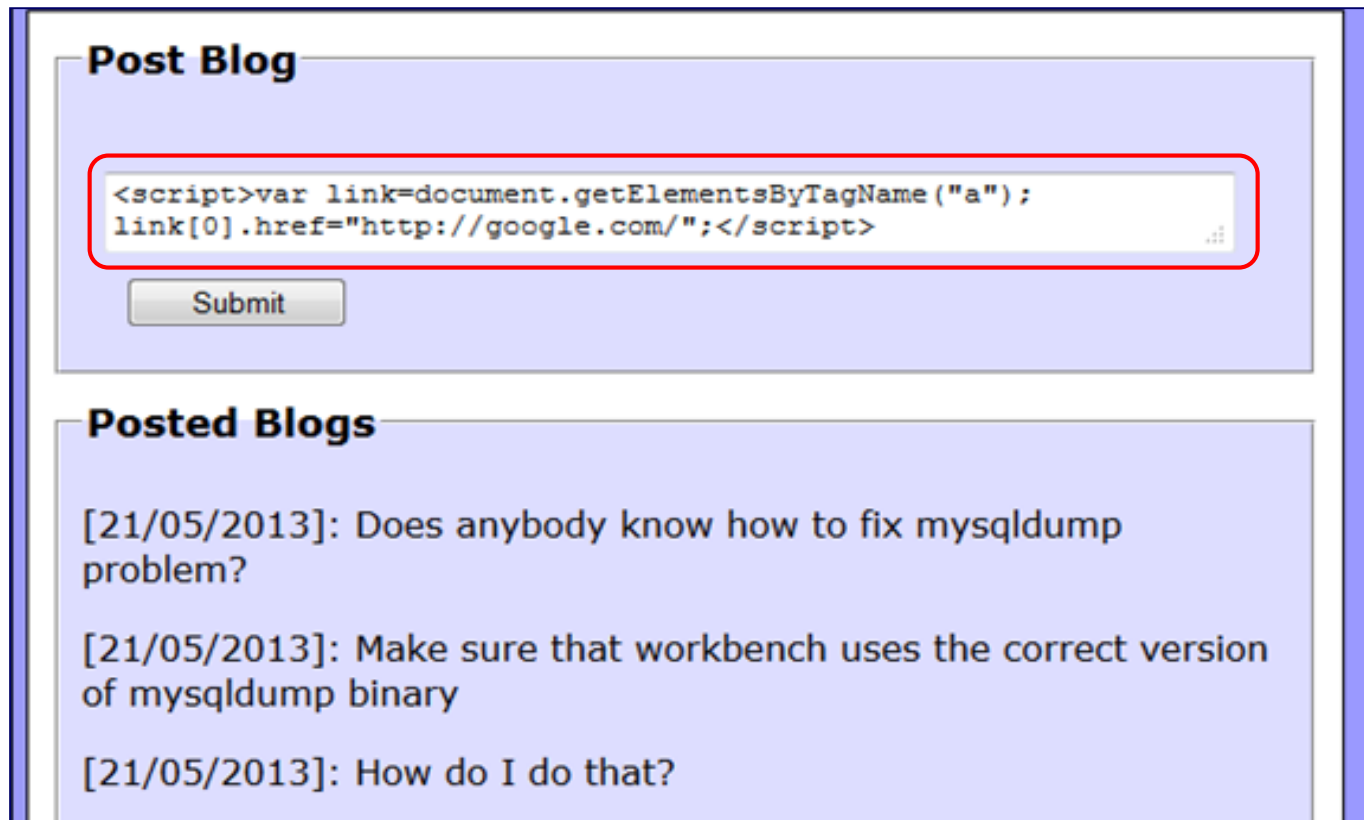
```
Run SQL query/queries on table customerdb.customer: ?  
1 SELECT *  
2 FROM customer  
3 WHERE customerID=1 or 1=1;
```



customerID	firstname	address	balance
1	Almira	Jl. Mawar No. 123, Surabaya	4500000
2	Baharudin	Jl. Mengkudu No. 456, Surabaya	1200000
3	Citra	Jl. Cendana No. 17, Surabaya	0
4	Derry	Jl. Delima No. 50, Surabaya	0
5	Erlina	Jl. Erlangga No. 44, Surabaya	0

# Script Injection

- Jangan secara langsung menampilkan masukan *user* yang berupa skrip → gunakan fungsi/*function* `htmlspecialchars`



The screenshot shows a web application interface. At the top is a 'Post Blog' section with a text input field and a 'Submit' button. The input field contains a JavaScript payload: `<script>var link=document.getElementsByTagName("a"); link[0].href="http://google.com/";</script>`. Below this is a 'Posted Blogs' section displaying a list of blog entries. Each entry starts with a date in brackets, followed by the text of the post.

**Post Blog**

```
<script>var link=document.getElementsByTagName("a");
link[0].href="http://google.com/";</script>
```

Submit

**Posted Blogs**

[21/05/2013]: Does anybody know how to fix mysqldump problem?

[21/05/2013]: Make sure that workbench uses the correct version of mysqldump binary

[21/05/2013]: How do I do that?

# XML Attack

- XML *attack* adalah *Denial-of-Service* (DoS) *attack*
- Merupakan XML schema yang *well-formed* dan *valid*

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
  <!ENTITY lol "lol">
  <!ENTITY lol2 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
  <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
  <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
  <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
  <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
  <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
  <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```



# Implementasi keamanan yang lemah

- Tidak mengimplementasikan sistem keamanan terbaru (*Web Browser, Web Server, ...*)
- Tidak mengimplementasikan mekanisme *password* yang baik
- Hanya mengandalkan validasi *client-side*
- Kecerobohan pengembang aplikasi *web* (*developer problem*)

# Developer Problem

- *Error reporting* tidak diaktifkan
- Variabel tanpa inisialisasi nilai

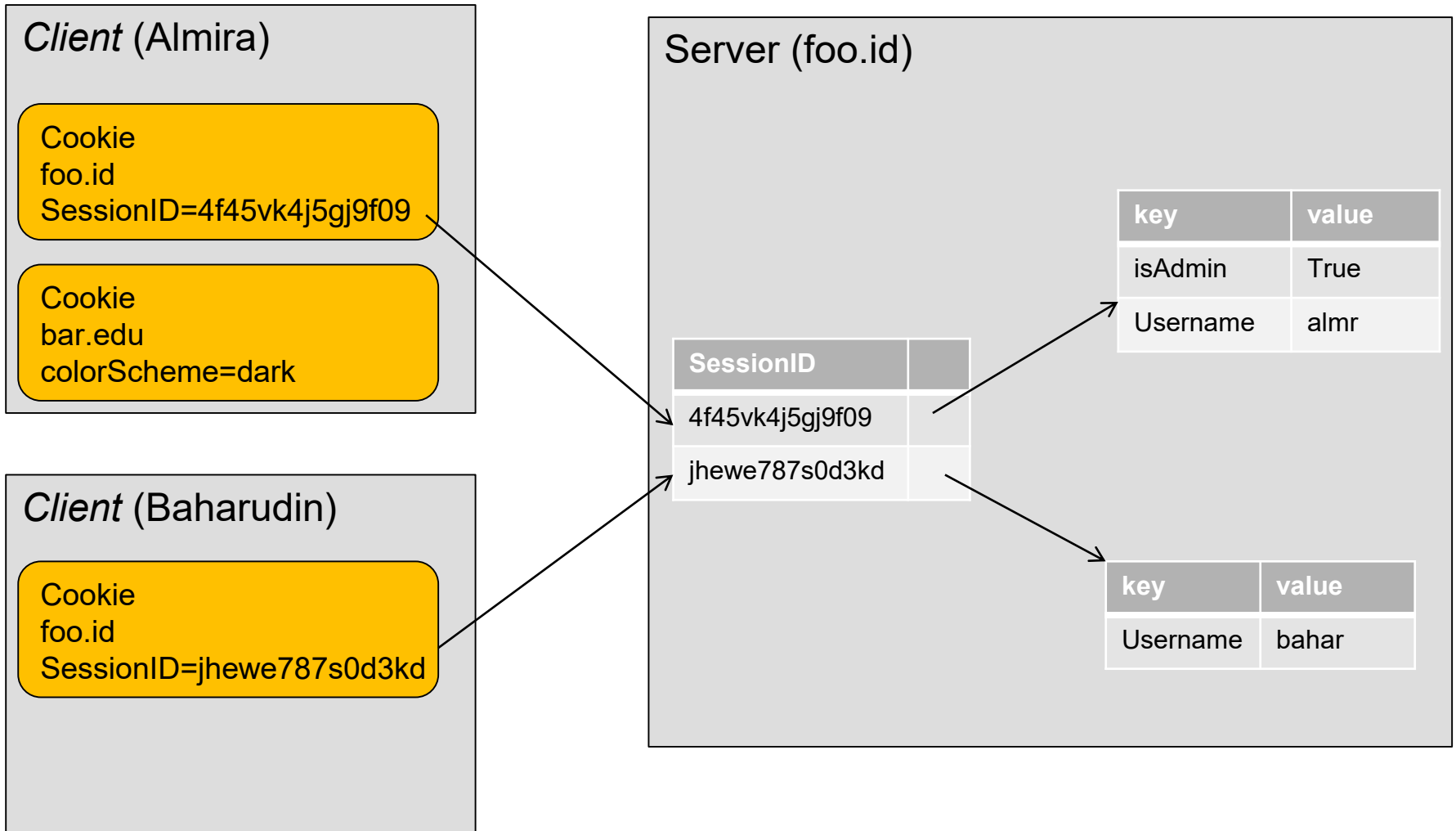
```
<?php
    $intNoValue ;
    if ($intNoValue == 0 )
        echo "<p>Equals Zero</p>";
?>
```

- Variabel global: `php.ini` → `register_global = On`

```
<legend>Login</legend>
<div class="field"> <!-- Name field and its error message -->
    <label for="username">Username:</label>
    <input type="text" name="strusername" id="username" />
</div>
<div class="field"> <!-- Password field and its error message -->
    <label for="password">Password:</label>
    <input type="password" name="strpassword" id="password" />
</div>
<div class="field">
    <input class="specialsubmit" type="submit" name="login" value="Login"/>
    <input class="specialsubmit" type="hidden" name="intOkay" value="1"/>
</div>
```

```
<?php
    if (checkUser($strusername, $strpassword))
        $intOkay = 1;
    if ($intOkay)
        echo "<p>Valid user</p>";
    function checkUser ($username, $password) {
        return false;
    }
?>
```

# Session



## Challenge #2

Apa persamaan dari **cookie** dan **session**?

Apa perbedaan dari **cookie** dan **session**?

## **Session untuk Authorization: Kategori halaman**

- Kategorikan halaman-halaman yang ada pada aplikasi *web* menjadi dua kelompok berikut:
  - Halaman **publik**, yaitu halaman yang dapat dilihat/diakses secara langsung oleh *user* tanpa melalui proses *login*
  - Halaman **non-publik**, yaitu halaman yang mengharuskan *user* untuk melakukan proses *login*

## Session untuk *Authorization*: Kategori halaman [2]

- Untuk halaman **non-publik**, tambahkan skrip seperti contoh berikut:

Private.php

```
<?php
    require 'adminPermission.inc';
?>

<!DOCTYPE html>
<html>
    <head>
```

adminPermission.inc

```
<?php
    session_start();
    if (!isset($_SESSION['isAdmin'])) // isAdmin adalah contoh label/nama session
    {
        // user akan diarahkan ke halaman login untuk authorization
        //header("Location: http://localhost/login.php");
        header("Location: http://".$_SERVER['HTTP_HOST'].'/login.php');
        exit();
    }
?>
```

## Challenge #3

Buatlah sebuah *file* PHP (halaman *private*/non-publik):

**Private.php**

(pastikan ada tampilan teks yang menandakan bahwa ini adalah halaman *private* atau non-publik)

Buatlah sebuah *file* INC:  
**adminPermission.inc**

Apa yang terjadi ketika *file* **Private.php** dibuka di *web browser*?

(*Hint*: Perhatikan URL setelah *file* **Private.php** dibuka)

# Session untuk Authorization: Halaman LOGIN

- Buat halaman **login.php** → halaman ini harus dikirimkan ke dirinya sendiri (cek konsep *self-submission*)

```
<?php
if (isset($_POST['login']))
{
    // lakukan validasi dan pemrosesan masukan data login di sini
}
?>

<!DOCTYPE html>
<html>
    <head>
        . . .
    </head>
    <body>
        <form action="login.php" method="POST">
            . . . // kotak isian masukan untuk username dan password
        </form>
    </body>
</html>
```

Cek penjelasan “**Validasi & pemrosesan masukan data login**” pada slide berikutnya

**Login**

Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

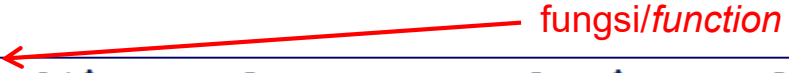


## Session untuk Authorization: Halaman LOGIN [2]

### Validasi & pemrosesan masukan data *login*:

- a. Cek apakah *username* dan *password* yang dimasukkan adalah benar. Lakukan pengecekan kondisi berikut:

```
if (checkPassword($_POST['username'], $_POST['password']))
```



- Di dalam fungsi **checkPassword**:
  - Gunakan **PDO Prepared Statement** untuk memvalidasi masukan data dengan yang ada di dalam basisdata. Contoh skrip SQL di dalam *statement prepare()*:

```
SELECT *  
FROM admins  
WHERE username = :username and password = SHA2(:password, 0)
```

- Ketika skrip SQL menghasilkan keluaran setidaknya satu baris, maka hasil validasi bernilai **true**

```
return $query->rowCount() > 0;
```

## Session untuk Authorization: Halaman LOGIN [3]

### Validasi & pemrosesan masukan data *login*:

- b. Jika pengecekan *username* dan *password* yang dimasukkan adalah benar (hasil pengecekan kondisi bernilai **true**)
- Aktifkan/buka session baru untuk mencatat bahwa seorang *user* “admin” telah berhasil *login* (label/nama *session* adalah **isAdmin**)

```
session_start();  
$_SESSION['isAdmin'] = true;
```

- Arahkan *user* “admin” ke halaman non-publik

```
header('Location: http://localhost/Private.php');
```

- *Exit* dari halaman Login

```
exit();
```

# Session untuk Authorization: Halaman LOGOUT

- Buat halaman **logout.php** untuk:
  - Membersihkan data *user* dari *session*:

```
session_start();  
unset($_SESSION['isAdmin']);
```

- Menampilkan informasi bahwa *user* telah berhasil *logout*

## Challenge #4

Buatlah halaman HOME (sebagai halaman publik) → nama file: **home.php**  
(pastikan ada tampilan teks yang menandakan bahwa ini adalah halaman *home*)

Buatlah halaman LOGIN → nama file:  
**login.php**

Buatlah halaman LOGOUT → nama file:  
**logout.php**

# Video Perkuliahan

- Penjelasan materi kuliah dapat dilihat via YouTube (*recording* Semester Gasal 2020/2021):  
[https://youtu.be/sUOJ4ZNP\\_s\\_M](https://youtu.be/sUOJ4ZNP_s_M)
- *Note: Kerjakan Challenge(s) berdasarkan slide Semester Gasal 2023/2024 (bukan berdasarkan slide yang ditampilkan dalam recording Semester Gasal 2020/2021)*



there is  
**ALWAYS**  
something to  
**BE GRATEFUL**  
for



whilehowsnapping.com

**SURROUND  
YOURSELF  
WITH POSITIVE  
PEOPLE WHO  
ARE GOING TO  
PUSH YOU  
TOWARD  
GREATNESS.**

PictureQuotes.com