LAPORAN

HASIL PERBAIKAN UJI KEAMANAN INFORMASI

WEB APLIKASI

bkpsdm-mesemku.surabaya.go.id/sijaka

PEMERINTAH KOTA SURABAYA

BADAN KEPEGAWAIAN & PENGEMBANGAN SUMBER DAYA MANUSIA

Berdasarkan hasil uji keamanan yang telah dilakukan oleh Kominfo pada Web Aplikasi bkpsdm-mesemku.surabaya.go.id. Ditemukan 48 vulnerability, 18 vulnerability dengan peringkat kritikal (critical), 27 vulnerability dengan peringkat tinggi (high), 1 vulnerability dengan peringkat sedang (medium), 2 vulnerabilityb dengan peringkat rendah (low).

Demikian perbaikan yang telah dilakukan oleh pihak BKPSDM terkait kemanan Web Aplikasi bkpsdm-mesemku.surabaya.go.id beserta Screenshoot sesuai hasil laporan dari pihak Kominfo. Berikut

Detail Kerentanan & Perbaikan

1. SQL Injection

SQL Injection adalah teknik penyalahgunaan celah keamanan pada lapisan database sebuah aplikasi. Bentuk penyerangan ini disalahgunakan sebagai bentuk ancaman berupa pencurian data melalui akses Database. Adapun sebagai langkah perbaikan dari ancaman SQL Injection tersebut adalah sebagai berikut.

```php
<?php
$id_tpp = htmlentities($_GET['id_tpp']);
$bulan_sekarang = date('m');
$query_get_data_bulan_ini = $db->prepare("SELECT user_master_asn.*
    ,beban_kerja_detail.*,beban_kerja_master.id_tpp as id_tpp,
    beban_kerja_master.tanggal_periode as tanggal_periode FROM
    user_master_asn left join beban_kerja_detail on user_master_asn.nip =
    beban_kerja_detail.nip left join beban_kerja_master on
    beban_kerja_master.id_tpp = beban_kerja_detail.id_tpp where
    beban_kerja_master.id_tpp = :id_tpp and user_master_asn.nip = :nip");
$query_get_data_bulan_ini->bindParam(':id_tpp',$id_tpp);
$query_get_data_bulan_ini->bindParam(':nip',$nip);
$query_get_data_bulan_ini->execute();
?>
```

Gambar 1.1

Gambar 1.1 adalah salah bentuk mengatasi celah keamanan SQL Injection, yaitu dengan penggunaan *Prepared Statements* membutuhkan 3 langkah: **Prepared, Bind,** dan **Execute**.

Pada proses pertama: **prepared**, kita mempersiapkan query yang akan dijalankan, tetapi tanpa penulisan variable pada setiap filternya. Bagian dimana data berada digantikan dengan tanda titik dua (:) dan diikuti dengan nama bebas.

Proses kedua adalah **bind**. Dalam tahap ini, kita akan mengirimkan data yang telah ditandai dalam proses *prepare*. Data disini adalah bagian yang diberi tanda titik dua (:) dan diikuti dengan nama bebas.

Setelah proses *prepare* dan *bind*, berikutnya adalah menjalankan prepared statement (**execute**).

2. Cross Site Scripting (XSS)

Cross site scripting adalah serangan injeksi kode pada sisi klien dengan menggunakan sarana halaman website atau web aplikasi. Cross site scripting ini sering digunakan untuk mencuri session cookies, yang memungkinkan penyerang untuk menyamar sebagai korban. Dengan cara inilah, peretas bisa mengetahui data-data sensitif milik korban. Adapun sebagai langkah perbaikan dari ancaman Cross Site Scripting (XSS) tersebut adalah sebagai berikut

```php
<?php
$id_tpp = htmlentities($_GET['id_tpp']);
$bulan_sekarang = date('m');
$query_get_data_bulan_ini = $db->prepare("SELECT user_master_asn.*
    ,beban_kerja_detail.*,beban_kerja_master.id_tpp as id_tpp,
    beban_kerja_master.tanggal_periode as tanggal_periode FROM
    user_master_asn left join beban_kerja_detail on user_master_asn.nip =
    beban_kerja_detail.nip left join beban_kerja_master on
    beban_kerja_master.id_tpp = beban_kerja_detail.id_tpp where
    beban_kerja_master.id_tpp = :id_tpp and user_master_asn.nip = :nip");
$query_get_data_bulan_ini->bindParam(':id_tpp',$id_tpp);
$query_get_data_bulan_ini->bindParam(':nip',$nip);
$query_get_data_bulan_ini->execute();
?>
```

Gambar 2.1

Gambar 2.1 adalah salah satu bentuk penggunaan htmlentities();. Fungsi ini digunakan untuk menghentikan proses penerjemahan tag HTML oleh browser. Fungsi lain dari penggunaan ini adalah untuk mengencode tag html menjadi karakter special.

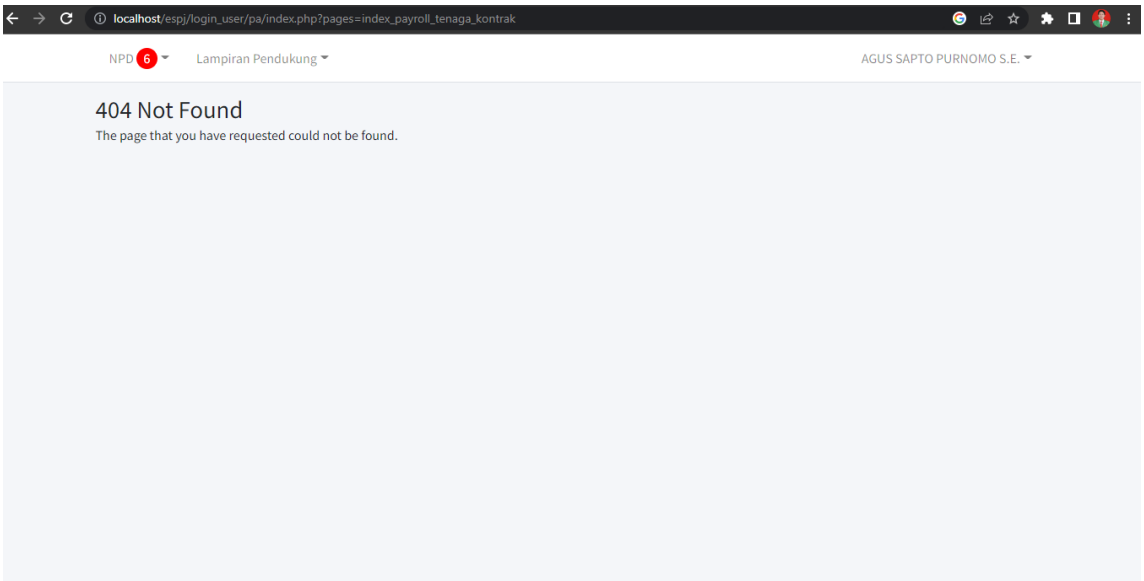
3. No Authentication Session Check

Celah Keamanan ini adalah celah keamanan yang terdeteksi karena tidak adanya pengecekan dalam session user role. Adapun cara penanganannya adalah berikut.

```
                                </nav>-->
                            </div>
                        </div>
                    </div>
                    <div class="sb-sidenav-footer">
                        <div class="small">Logged in as:</div>
                        <b><?php echo $penyedia['nama_penyedia']; ?></b>
                    </div>
                </nav>
            </div>
            <?php
                $actual_link = "http://$_SERVER[REQUEST_URI]";
                if (strrpos($actual_link, $_SESSION['role']) !== false){ //cek path role sesuai login_user
                if(@$_GET['pages'] == ''){ ?>
                    <div id="layoutSidenav_content">
                        <main>
                            <div class="container-fluid">
                                <h1 class="mt-4">Dashboard</h1>
                                <ol class="breadcrumb mb-4">
                                    <li class="breadcrumb-item active">Dashboard</li>
                                </ol>
                                <div class="card mb-4">
```

3.1. Fungsi Pengecekan User Session Role

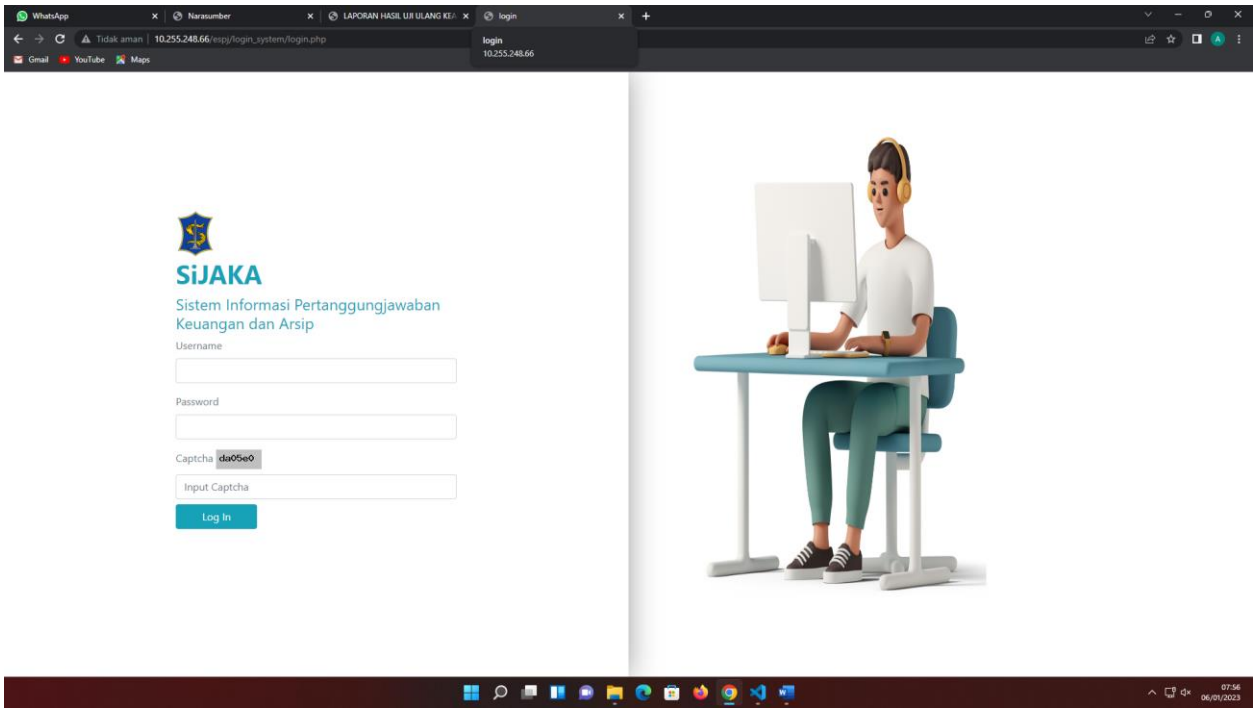
Gambar 4.1. Screenshoot diatas adalah salah satu bentuk pembatasan session role user, adapun yang dimaksud adalah agar mencegah pembatasan apabila url link dicopy paste pada browser.



3.2. 404 Not Found Session Beda User

Gambar 3.2. Berikut adalah Notifikasi ketika Session Role User dirubah secara manual pada URL. Ketika session_destroy(): masih belum dilakukan dan role user session dirubah, hasil form akan tidak ditemukan atau 404 Not Found.

4. Login Page Password-Guessing Attack



Gambar 4.1

Gambar 4.1. Pada halaman login user dilakukan menginput captcha yang sudah disediakan untuk menghindari brute force.

5. User Enumeration

Celah keamanan ini adalah celah dimana penyerang bisa mengetahui username yang terdaftar di dalam aplikasi. Hal ini bisa didapatkan karena web aplikasi membedakan notifikasi antara username yang belum terdaftar dan juga username yang sudah terdaftar. Cara mengatasi masalah ini adalah menyamakan notifikasi.

```
                $_SESSION['nik'] = $check_pass_tenaga_kontrak['nik'];
                header('Location: ../../tenaga_kontrak/tenaga_kontrak_approve/index.php');
            }
            else{
                $percobaan = htmlentities($_POST['percobaan']);
                header('Location: ../../login_system/login.php?percobaan='.$percobaan.'&status=password_salah');
            }
        }
    }
    else{
            //echo "<script>alert('username belum terdaftar');history.go(-1)</script>";
        $percobaan = htmlentities($_POST['percobaan']);
        header('Location: ../../login_system/login.php?percobaan='.$percobaan.'&status=password_salah');
    }

    }else{
        echo "<script>alert('Login gagal! Capctha tidak sesuai!')</script>";
        header('Location: ../../login_system/login.php?percobaan='.$percobaan.'&status=captcha_salah');
    }

}
?>
```

Gambar 5.1

```
105         </div>
106         <script src="../asset/assets/js/jquery.min.js"></script>
107         <script src="assets/bootstrap/js/bootstrap.min.js"></script>
108         <?php
109             if(htmlentities(isset($_GET['status']))){
110                 $status = htmlentities($_GET['status']);
111
112                 if($status == 'password_salah'){
113         ?>
114         <script>alert('Username atau Password salah');</script>
115         <?php } elseif($status == 'captcha_salah'){?>
116         <script>alert('Login gagal! Capctha tidak sesuai!');</script>
117         <?php }}?>
118     </body>
119     <script>
120     var sec = 15;
```

Gambar 5.1

Gambar 5.2 menunjukan lemparan untuk username terdaftar dan tidak terdaftar adalah sama. Gambar 5.2 menunjukan notifikasi yang sama.