

# SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC002024186949, 17 September 2024

## Pencipta

Nama : **Dr. Fatwa Tentama, S.Psi., M.Si., Herman Yuliansyah, S.T., M. Eng., Ph.D. dkk**  
Alamat : Sidorejo No 002, RT 006, Ngestiharjo, Kasihan, Bantul, DI Yogyakarta, 55182  
Kewarganegaraan : Indonesia

## Pemegang Hak Cipta

Nama : **Dr. Fatwa Tentama, S.Psi., M.Si., Herman Yuliansyah, S.T., M. Eng., Ph.D. dkk**  
Alamat : Sidorejo No 002, RT 006, Ngestiharjo, Kasihan, Bantul, DI Yogyakarta, 55182  
Kewarganegaraan : Indonesia  
Jenis Ciptaan : **Program Komputer**  
Judul Ciptaan : **Aplikasi Early Detection Work Readiness Berbasis Employability Skill (EDWRES)**  
Tanggal dan tempat diumumkan untuk pertama kali : 16 September 2024, di Yogyakarta  
di wilayah Indonesia atau di luar wilayah Indonesia  
Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.  
Nomor pencatatan : 000759428

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA  
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL  
u.b

Direktur Hak Cipta dan Desain Industri

IGNATIUS M.T. SILALAH  
NIP. 196812301996031001

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

**LAMPIRAN PENCIPTA**

No	Nama	Alamat
1	Dr. Fatwa Tentama, S.Psi., M.Si.	Sidorejo No 002, RT 006, Ngestiharjo, Kasihan, Bantul
2	Herman Yuliansyah, S.T., M. Eng., Ph.D.	Perumahan Graha Nirmala F17, Jeblog, Tirtonirmolo, Kasihan, Bantul
3	Zahro Varisna Rohmadani, S.Psi., M.Psi., Psikolog.	Perum Palem Indah A.1 RT 06 RW 27 Nogosari, Sidokarto, Godean, Sleman
4	Dr. Surahma Asti Mulasari, S.Si., M.Kes.	Perumahan Sidorejo Gang Nakula No. 161 Ngestiharjo, Kasihan, Bantul
5	Lu'lu' Nafiati, S.E., M.Sc.	Dongkelan RT 02 No 39 Panggungharjo, Sewon, Bantul
6	Sulistyawati, S.Si., M.PH., Ph.D.	Sepat, 005/002, Ngoro-oro, Patuk, Patuk, Gunung Kidul
7	Dr. Tri Wahyuni Sukesni, S.Si., M.PH.	Dusun Lojajar RT 003 RW 022 Sinduharjo, Ngaglik, Sleman

**LAMPIRAN PEMEGANG**

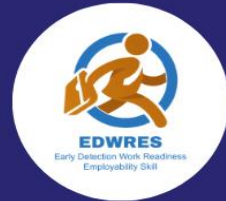
No	Nama	Alamat
1	Dr. Fatwa Tentama, S.Psi., M.Si.	Sidorejo No 002, RT 006, Ngestiharjo, Kasihan, Bantul
2	Herman Yuliansyah, S.T., M. Eng., Ph.D.	Perumahan Graha Nirmala F17, Jeblog, Tirtonirmolo, Kasihan, Bantul
3	Zahro Varisna Rohmadani, S.Psi., M.Psi., Psikolog.	Perum Palem Indah A.1 RT 06 RW 27 Nogosari, Sidokarto, Godean, Sleman
4	Dr. Surahma Asti Mulasari, S.Si., M.Kes.	Perumahan Sidorejo Gang Nakula No. 161 Ngestiharjo, Kasihan, Bantul
5	Lu'lu' Nafiati, S.E., M.Sc.	Dongkelan RT 02 No 39 Panggungharjo, Sewon, Bantul
6	Sulistyawati, S.Si., M.PH., Ph.D.	Sepat, 005/002, Ngoro-oro, Patuk, Patuk, Gunung Kidul
7	Dr. Tri Wahyuni Sukesni, S.Si., M.PH.	Dusun Lojajar RT 003 RW 022 Sinduharjo, Ngaglik, Sleman







# 2024



# Work Readiness

Pendekatan empirik untuk mengidentifikasi secara dini kesiapan kerja lulusan baik lulusan Sekolah Menengah Kejuruan atau lulusan Perguruan Tinggi sebelum memasuki dunia kerja.

Tujuan utama dari deteksi dini kesiapan kerja adalah untuk mendiagnosis dan mengevaluasi

- • pengetahuan, ketrampilan, kompetensi dan
- • mentalitas yang diperlukan agar berhasil dalam
- • mendapatkan dan menjalankan pekerjaannya.



Universitas Ahmad Dahlan



## Profil Work Readiness

Work Readiness adalah pendekatan atau metode untuk mengidentifikasi secara dini kesiapan seseorang, terutama pelajar atau lulusan baru, untuk memasuki dunia kerja. Tujuan utama dari deteksi dini ini adalah untuk mengevaluasi apakah individu memiliki keterampilan, kompetensi, dan mentalitas yang diperlukan untuk berhasil dalam lingkungan kerja profesional. Dengan menggunakan Early Detection Work Readiness, perusahaan atau lembaga pendidikan dapat mempersiapkan individu agar lebih siap dan kompetitif dalam pasar kerja.

Aplikasi work readiness dibuat untuk platform Android dengan beberapa fitur yang dimiliki antara lain :

a. Konsultasi

Aplikasi akan menampilkan daftar item yang dapat dipilih oleh pengguna. Setelah pengguna memilih item dari daftar, aplikasi akan menampilkan hasil deteksi yang relevan dengan item tersebut. Aplikasi memberikan opsi kepada pengguna untuk mencetak hasil deteksi dalam format PDF yang dapat disimpan.

b. Login dan Registrasi

Pakar dan user dapat melakukan login di aplikasi ini untuk melakukan konsultasi dan menambahkan data data pakar. Dan user dapat melakukan registrasi terlebih dahulu jika belum memiliki akun untuk melakukan registrasi.

c. Pakar

Pakar dapat menambahkan data data yang dibutuhkan untuk user dapat melakukan konsultasi . datanya berupa eas, indikator , tips dan pengetahuan.

## Mekanisme Penggunaan Aplikasi Work Readiness

Untuk menggunakan Aplikasi Work Readiness langkah langkah nya adalah sebagai berikut :

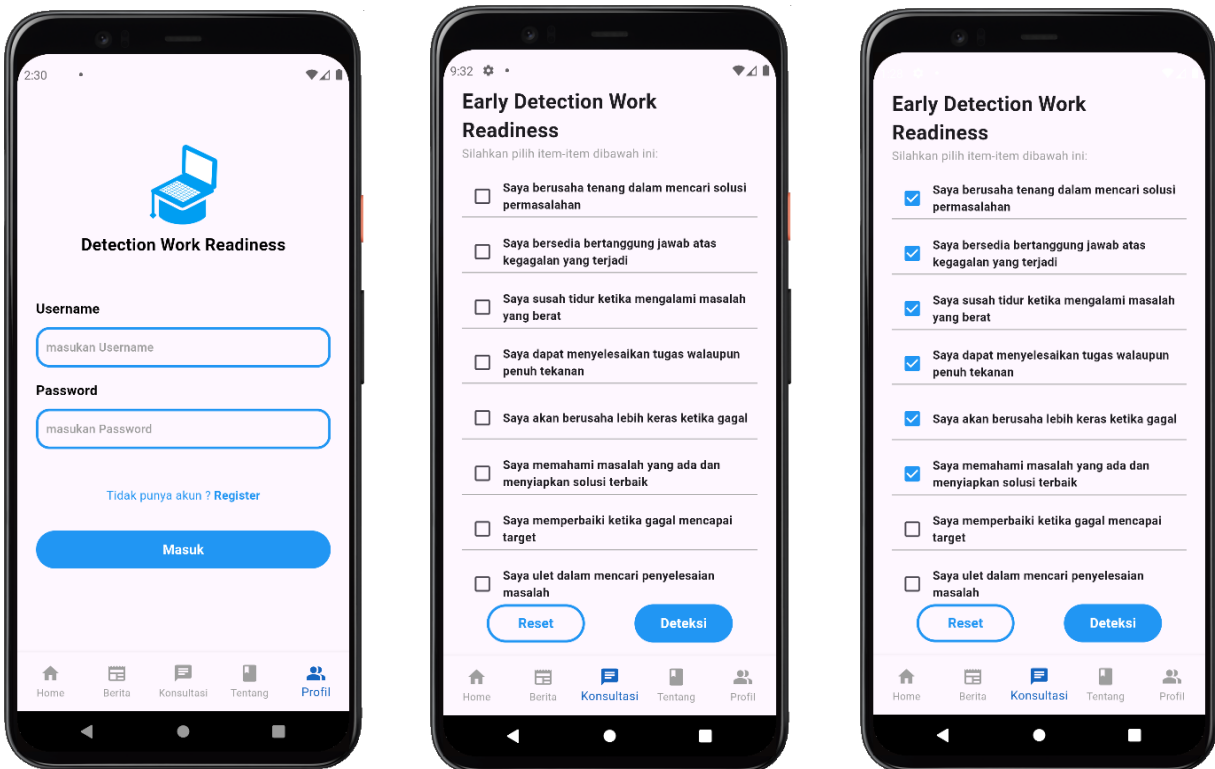
A. User

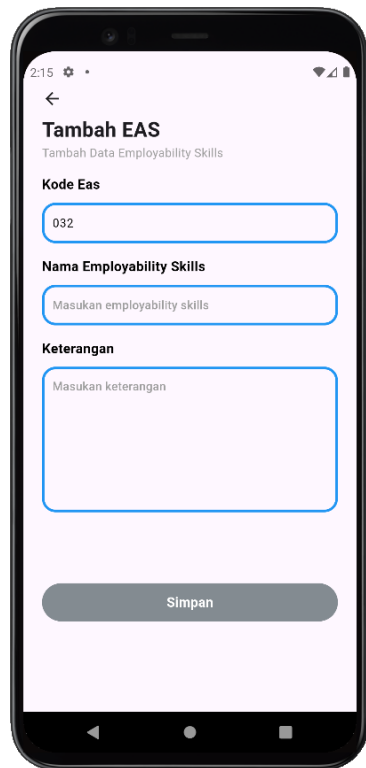
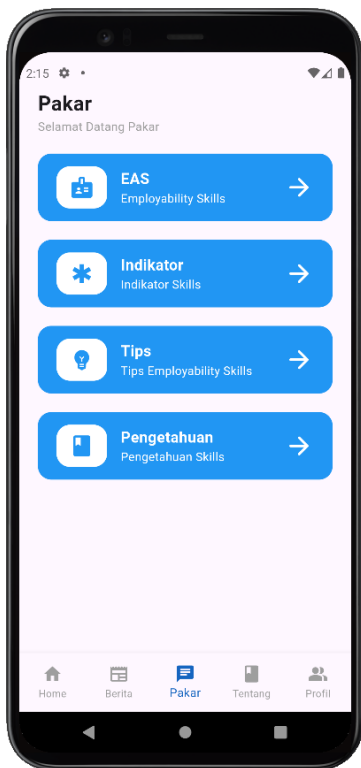
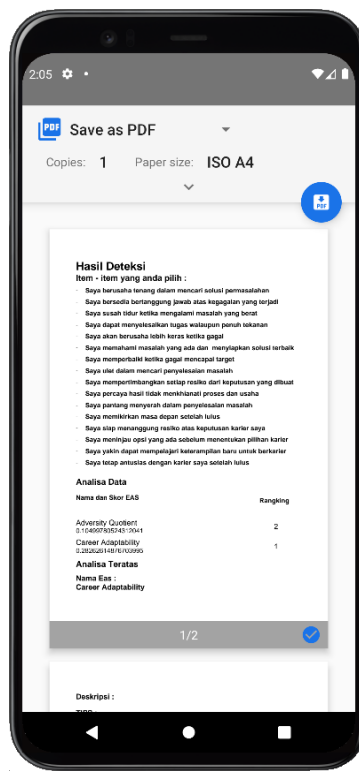
- a. Download dan Install aplikasi dari link yang disediakan.
- b. Melakukan Registrasi terlebih dahulu jika belum mempunyai akun.
- c. Setelah memiliki akun dan login anda dapat melakukan konsultasi.
- d. Pilih item yang sesuai dengan kesiapan yang anda rasakan untuk memasuki dunia kerja.
- e. Setelah memilih item anda dapat melakukan fungsi deteksi.
- f. Dari hasil deteksi anda dapat melihat kesiapan kerja ada dan diberikan tips untuk dapat lebih siap lagi dalam memasuki dunia kerja.
- g. Dari hasil deteksi atau diagnosa anda dapat menyimpan data tersebut.

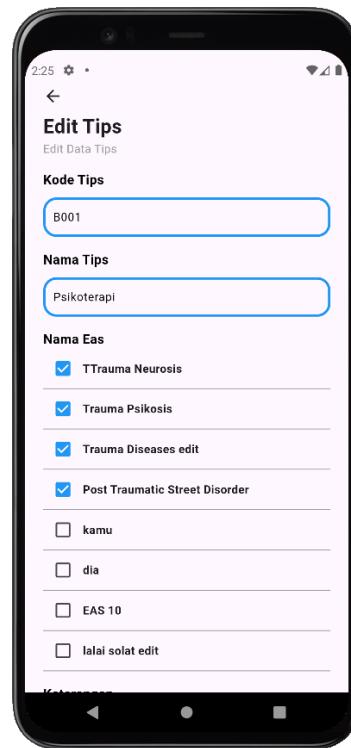
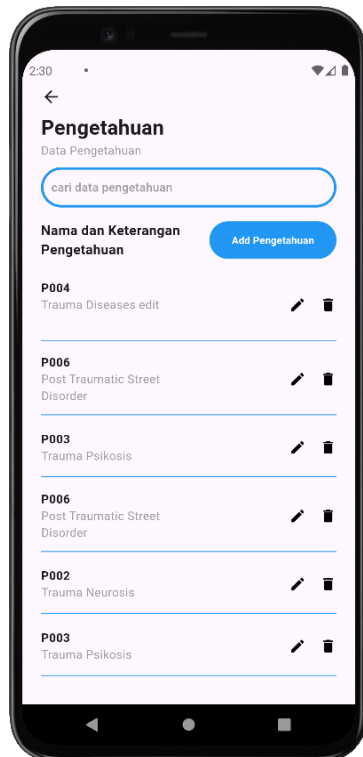
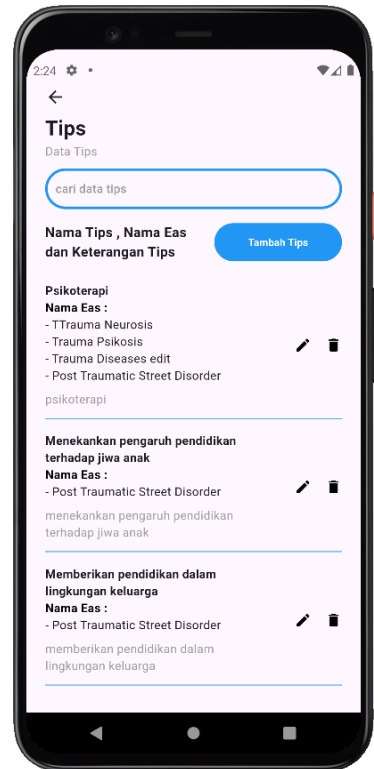
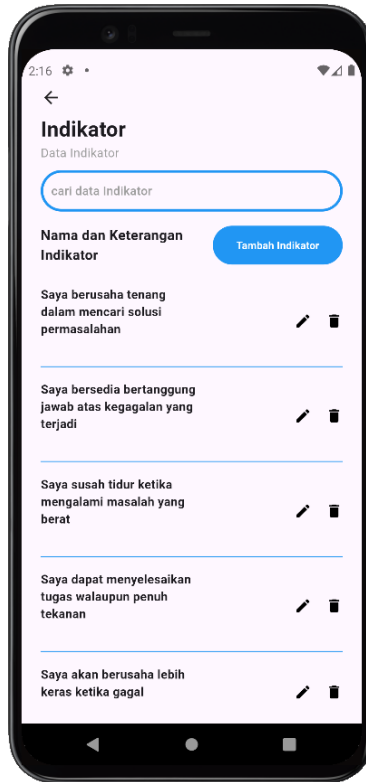
## B. Pakar

- Download dan Install aplikasi dari link yang disediakan.
- Pakar mendapatkan akun yang di buat admin.
- Setelah memiliki akun dan login pakar dapat menambahkan data data yang dibutuhkan user untuk melakukan konsultasi, seperti data eas, indikator, tips dan pengetahuan.
- Pakar juga dapat menyesuaikan data data tersebut seperti , menambahkan data, edit data hapus data.

### Gambaran Aplikasi Work Readiness







## Source Code Work Readiness

### Login dan Register

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:work_readiness/model/login_user.dart';

class LoginServices {
  final String apiUrl =

    "https://detectionworkreadiness.com/Rest_api/login_user.php";

  Future<LoginUser> login(String username, String password) async {
    final response = await http.post(
      Uri.parse(apiUrl),
      body: {
        'username': username,
        'password': password,
      },
    );

    // print(response);
    if (response.statusCode == 200) {
      // Jika respon berhasil, parsing JSON
      return LoginUser.fromJson(jsonDecode(response.body));
    } else {
      // Jika gagal, lempar exception
      throw Exception('Failed to login');
    }
  }
}

<?php
// Set header untuk mengembalikan response dalam format JSON
header('Content-Type: application/json');

// Menghubungkan ke database
```



```

include 'dbconnection.php'; // Pastikan Anda memiliki file config.php untuk koneksi ke database
$con = dbconnection();
// print "<p>Study PHP at " . $con . "</p>";

// Mendapatkan data dari request body
// $data = json_decode(file_get_contents("php://input"), true);
$username = $_POST["username"];
$password = $_POST["password"];
// Validasi input
if ($username && $password) {
    $username = mysqli_real_escape_string($GLOBALS['koneksi'], $username);
    $password = md5(mysqli_real_escape_string($GLOBALS['koneksi'], $password));

    $query = "SELECT * FROM user WHERE username = '$username' AND password = '$password' AND level != 'admin'";
    $result = mysqli_query($GLOBALS['koneksi'], $query);

    if ($result) {
        $num_result = mysqli_num_rows($result);

        if ($num_result > 0) {
            $user = mysqli_fetch_assoc($result);

            // Siapkan data yang akan dikirimkan dalam response
            $response = [
                'status' => 'success',
                'message' => 'Login berhasil',
                'user' => [
                    'id' => $user['id_user'],
                    'username' => $user['username'],
                    'level' => $user['level']
                ]
            ];
        } else {
            // Jika login gagal
            $response = [
                'status' => 'error',
                'message' => 'Username atau password salah, atau level tidak diperbolehkan'
            ];
        }
    }
}

```

```

    } else {
        // Jika query gagal
        $response = [
            'status' => 'error',
            'message' => 'Terjadi kesalahan pada server'
        ];
    }
} else {
    // Jika input tidak valid
    $response = [
        'status' => 'error',
        'message' => 'Data username dan password harus diisi'
    ];
}

// Mengirimkan response dalam format JSON
echo json_encode($response);
?>

```

## **Konsultasi**

```

// Method untuk melakukan POST request
Future<void> postHasilDiagnosa() async {
    setState() {
        isLoading = true;
    });
    SharedPreferences prefs = await SharedPreferences.getInstance();
    var idUser = prefs.getString('id_user');
    print(idUser);
    final url = Uri.parse(
        // 'https://detectionworkreadiness.com/Rest_api/hasil_diagnosa.php'
        'https://detectionworkreadiness.com/Rest_api/hasil_diagnosa.php');

    // Mempersiapkan form data dengan array
    Map<String, String> formData = {};
    for (int i = 0; i < widget.kodeIndikatorArray.length; i++) {
        formData['kode_indikator[$i]'] = widget.kodeIndikatorArray[i];
    }
    formData['id_user'] = idUser.toString();
}

```

```

try {
    final response = await http.post(
        url,
        body: formData,
        headers: {
            'Content-Type': 'application/x-www-form-urlencoded',
        },
    );

    if (response.statusCode == 200) {
        var res = json.decode(response.body);
        print(res.toString());
        if (res != 'Data harus lebih dari 1') {
            final Map<String, dynamic> data = json.decode(response.body);
            var jumdataeas = data['dataEas'].length;
            var datadia = data['diagnosa'];
            var dataeas = data['dataEas'];
            print(datadia.toString());

            //low rengking

            String? keyWithLowestRanking;
            int? lowestRanking;

            // Iterasi data untuk mencari key dengan ranking terbesar
            datadia.forEach((key, value) {
                int currentRanking = value['ranking'];

                if (lowestRanking == null || currentRanking > lowestRanking!) {
                    lowestRanking = currentRanking;
                    keyWithLowestRanking = key;
                }
            });

            // Mencari nama_eas berdasarkan kode_eas
            String? namaEas;
            for (var item in dataeas) {
                if (item['eas']['kode_eas'] == keyWithLowestRanking) {
                    namaEas = item['eas']['nama_eas'];
                    break;
                }
            }
        }
    }
}

```

```

    }
}

print(keyWithLowestRanking.toString());
print(namaEas.toString());
setState() {
    isLoading = false;
    dataDeteksi = data;
    jmlEas = jumdataeas;
    lastData = jumdataeas - 1;
    lowEas = namaEas.toString();
});
} else {
    setState() {
        isLoading = false;
    });
}
} else {
    setState() {
        isLoading = false;
    });
    print('Failed to post data. Status code: ${response.statusCode}');
}
} catch (e) {
    setState() {
        isLoading = false;
    });
    print('Error: $e');
}
}

```

```
<?php
```

```
header('Content-Type: application/json');
```

```
// Koneksi ke database
```

```
include 'dbconnection.php'; // Pastikan Anda memiliki file config.php untuk koneksi ke
database
```

```
$con = dbconnection();
```



```

$kodeindikator = $_POST["kode_indikator"];
$idUser = $_POST["id_user"];

// Memeriksa apakah 'kode_indikator' ada di POST request
if ($kodeindikator && $idUser) {
    // Mengambil data 'kode_indikator' dari POST request
    $kodeindikator = $_POST["kode_indikator"];

    // Memeriksa apakah $kodeindikator adalah array
    if (is_array($kodeindikator)) {
        $jml_indikator = count($kodeindikator);
    } else {
        $jml_indikator = 0; // Jika bukan array, set jumlah indikator ke 0
    }
} else {
    $jml_indikator = 0; // Jika 'kode_indikator' tidak ada, set jumlah indikator ke 0
}

if ($jml_indikator > 1) {
    // Ambil id_user dari sesi
    $id_user = $idUser;

// Insert ke tabel konsultasi
$insertQuery = "INSERT INTO konsultasi VALUES (',$id_user',now())";
$stmt = mysqli_prepare($GLOBALS['koneksi'], $insertQuery);

$data = [];
$hasil = [];
$ranking = [];
$array_eas = [];
$datatips = [];
$namaIndikatorArray = [];
// Pastikan POST['kode_indikator'] ada dan merupakan array
if (isset($_POST['kode_indikator']) && is_array($_POST['kode_indikator'])) {

    foreach ($_POST['kode_indikator'] as $kode_indikator) {

        $result = mysqli_query($GLOBALS['koneksi'],

```

```
"SELECT kode_eas,mb,md FROM pengetahuan WHERE kode_indikator =
'$kode_indikator'");
```

```
$num_result[$kode_indikator] = $result->num_rows;
```

```
while($rows = mysqli_fetch_array($result,MYSQLI_ASSOC))
```

```
{
```

```
    $data[$rows['kode_eas']][]=$rows;
```

```
}
```

```
// $cobaEas = to_eas();
```

```
// echo $data[$rows['kode_eas']];
```

```
}
```

```
foreach ($_POST['kode_indikator'] as $kode_indikator) {
```

```
    $result = mysqli_query($GLOBALS['koneksi'], "SELECT nama_indikator FROM indikator
    WHERE kode_indikator = '$kode_indikator'");
```

```
    $rows = mysqli_fetch_array($result,MYSQLI_ASSOC);
```

```
    $namaIndikatorArray[] = $rows['nama_indikator'];
```

```
}
```

```
foreach ($data as $kode_eas => $pengetahuan) {
```

```
    $jml = count($pengetahuan);
```

```
    if ($jml >= 1) { // Process even if there's only one item
```

```
        $mb = $pengetahuan[0]['mb'];
```

```
        $md = $pengetahuan[0]['md'];
```

```
        for ($i = 1; $i < $jml; $i++) {
```

```
            $mb = rumus($pengetahuan[$i]['mb'], $mb);
```

```
            $md = rumus($pengetahuan[$i]['md'], $md);
```

```

    }

    $hasil[$kode_eas]['mb'] = $mb;
    $hasil[$kode_eas]['md'] = $md;
    $hasil[$kode_eas]['CF'] = $mb - $md;

    unset($mb, $md);
}
// echo $kode_eas;
}

if (!empty($hasil)) { // jika ada hasil

    // Menambahkan ranking
    foreach ($hasil as $kode_eas => $pengetahuan) {
        $ranking[$kode_eas] = $pengetahuan['CF'];
    }

    arsort($ranking);

    $r = 1;
    foreach ($ranking as $kode_eas => $CF) {
        $hasil[$kode_eas]['ranking'] = $r;
        $r++;
    }

    foreach ($hasil as $kode_eas => $pengetahuan) {

        $result = mysqli_query($GLOBALS['koneksi'], "SELECT * FROM eas
        WHERE kode_eas = '$kode_eas'");

        $seas = mysqli_fetch_array($result, MYSQLI_ASSOC);
        // Menambahkan elemen baru ke array
        $array_eas[] = [
            // 'kode_eas' => $kode_eas,
            // 'pengetahuan' => $pengetahuan,
            'eas' => $seas,
        ];
    }
}

```

```

$data = [];

$result = mysqli_query($GLOBALS['koneksi'], "SELECT * FROM tips
JOIN data_tips ON tips.kode_tips = data_tips.kode_tips
WHERE kode_eas = '$kode_eas'");

$num_result = $result->num_rows;

if ($num_result > 0) {

    while ($rows = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
        // Menambahkan setiap hasil ke array $data
        $datatips[] = $rows;
    }
}

$data = [];
$data['diagnosa'] = $hasil;
$data['dataEas'] = $array_eas;
$data['dataTips'] = $datatips;
$data['namaIndikator'] = $namaIndikatorArray;
// $dataEas['dataEas'] =

// Mencari nilai terbesar
$data['nilai_terbesar'] = max($ranking);

// Mencari kode_eas dari nilai terbesar
$data['kode_eas'] = array_search($data['nilai_terbesar'], $ranking);
}

// Menampilkan hasil dalam format JSON
echo json_encode($data, JSON_PRETTY_PRINT);

}
} else {
    echo json_encode('Data harus lebih dari 1');
}
}

```



```

function rumus($nilai, $nilai2){

    $nilai = (float)$nilai;

    $nilai2 = (float)$nilai2;

    return $nilai + ($nilai2 * (1 - $nilai));

}

function to_eas($kode_eas){

    $result = mysqli_query($GLOBALS['koneksi'], "SELECT * FROM eas
        WHERE kode_eas = '$kode_eas'");

    $eas = mysqli_fetch_array($result, MYSQLI_ASSOC);

    return $eas;

}

function to_tips($kode_eas){

    $result = mysqli_query($GLOBALS['koneksi'], "SELECT * FROM tips
        JOIN data_tips ON tips.kode_tips = data_tips.kode_tips
        WHERE kode_eas = '$kode_eas'");

    $num_result = $result->num_rows;

    if($num_result > 0)
    {

        while($rows = mysqli_fetch_array($result, MYSQLI_ASSOC))

        {

            $data[]=$rows;

        }

    }

}

```

```

        return $data;
    }
}

function to_nama_indikator($kode_indikator){

    $result = mysqli_query($GLOBALS['koneksi'], "SELECT nama_indikator FROM indikator
        WHERE kode_indikator = '$kode_indikator'");

    $rows = mysqli_fetch_array($result,MYSQLI_ASSOC);

    return $rows['nama_indikator'];

}

```

?>

## **Pakar**

```

Future<void> fetchEasData() async {
    final response = await http.get(
        Uri.parse('https://detectionworkreadiness.com/Rest_api/list_eas.php'));

    if (response.statusCode == 200) {
        final data = json.decode(response.body);
        final List eas = data['eas'];

        setState() {
            easList = eas.map((json) => Eas.fromJson(json)).toList();
            isLoading = false;
        });
    } else {
        throw Exception('Failed to load data');
    }
}

```

```
<?php
// Header untuk mengatur response sebagai JSON
header('Content-Type: application/json');

// Koneksi ke database
include 'dbconnection.php'; // Pastikan Anda memiliki file config.php untuk koneksi ke database
$con = dbconnection();

$limit = 20;

if (isset($_GET["page"])) {

    $page = $_GET["page"];

} else {

    $page = 1;

}

$start_from = ($page-1) * $limit;

if (isset($_GET['src'])) { //hasil pencarian

    $src = escape_string($_GET['src']);

    $result = mysqli_query($GLOBALS['koneksi'],
        "SELECT * FROM eas WHERE kode_eas LIKE '%$src%'
        OR nama_eas LIKE '%$src%' OR keterangan LIKE '%$src%'");

} else {

    $result = mysqli_query($GLOBALS['koneksi'],
        "SELECT * FROM eas LIMIT $start_from, $limit");

}

$total_result = mysqli_query($GLOBALS['koneksi'],
    "SELECT kode_eas FROM eas");
```

```

$num_result = $result->num_rows;

if($num_result > 0)
{
    while($rows = mysqli_fetch_array($result,MYSQLI_ASSOC))
    {
        $data['eas'][]=$rows;
    }
}

$data['limit'] = $limit;
$data['current_page'] = $page;
$data['total_records'] = $total_result->num_rows;

// Mengembalikan data dalam format JSON
echo json_encode($data);

```

?>

```

Future<void> fetchIndikatorData() async {
    final response = await http.get(Uri.parse(
        'https://detectionworkreadiness.com/Rest_api/get_indikator.php'));

    if (response.statusCode == 200) {
        final data = json.decode(response.body);
        final List indikatorData = data['indikator'];

        setState() {
            indikatorList =
                indikatorData.map((json) => Indikator.fromJson(json)).toList();
            isLoading = false;
        });
    } else {
        throw Exception('Failed to load data');
    }
}

```



```
}
```

```
<?php
```

```
// Header untuk mengatur response sebagai JSON
```

```
header('Content-Type: application/json');
```

```
// Koneksi ke database
```

```
include 'dbconnection.php'; // Pastikan Anda memiliki file config.php untuk koneksi ke database
```

```
$con = dbconnection();
```

```
$limit = 1000;
```

```
    // if (isset($_GET["page"])) {
```

```
        //     $page = $_GET["page"];
```

```
    // }else{
```

```
        //     $page = 1;
```

```
    // }
```

```
$page = 1;
```

```
$start_from = ($page-1) * $limit;
```

```
// if (isset($_GET['src'])) { //hasil pencarian
```

```
    //     $src = escape_string($_GET['src']);
```

```
    //     $result = mysqli_query($GLOBALS['koneksi'],
```

```
    //         "SELECT * FROM indikator WHERE kode_indikator LIKE '%$src%'
```

```
    //         OR nama_indikator LIKE '%$src%' OR keterangan LIKE '%$src%'");
```

```
    // }else{
```

```
    //     $result = mysqli_query($GLOBALS['koneksi'],
```

```
    //         "SELECT * FROM indikator LIMIT $start_from, $limit");
```

```
    // }
```

```
$result = mysqli_query($GLOBALS['koneksi'],
"SELECT * FROM indikator LIMIT $start_from, $limit");

$total_result = mysqli_query($GLOBALS['koneksi'],
    "SELECT kode_indikator FROM indikator");

$num_result = $result->num_rows;

if($num_result > 0)
{
    while($rows = mysqli_fetch_array($result,MYSQLI_ASSOC))
    {
        $data['indikator'][]=$rows;
    }
}

$data['limit']          = $limit;
$data['current_page'] = $page;
$data['total_records'] = $total_result->num_rows;

// view('indikator/list_indikator',$data);
echo json_encode($data);
```

?>