

Junior Web Developer

Author : Rizki Fajar Purnomo, Politeknik Negeri Cilacap, 29 Juli 2025

Software & Tools :

- Xampp – *framework* web dan *database* server
- Visual Studio Code – *software* teks editor
- Google Chrome – web browser untuk menampilkan output
- GitHub – untuk menyimpan dan akses kode
- Laptop – untuk ngoding
- Bootstrap – *framework* untuk *styling*

Step-by-step :

Membuat file index.php

Code :

```
1 <!-- index.php -->
2 <!--
3   Author : Rizki Fajar Purnomo
4   Version : 1.0
5 -->
6 <html>
7
8 <head>
9 <title>ISP 3M0</title>
10 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-LnV7dVzj6uS2u38Kp9M/r118PCWtYK832p0p0pK2C1uLupj697PH4R/5mcr" crossorigin="anonymous">
11 </head>
12 <body>
13 <header>
14 <!-- untuk bar navigasi -->
15 <nav class="navbar navbar-expand-lg bg-body-tertiary">
16 <div class="container-fluid">
17 <a class="navbar-brand" href="index.php">BEASISWA</a>
18 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
19 <span class="navbar-toggler-icon"></span>
20 </button>
21 <div class="collapse navbar-collapse" id="navbarSupportedContent">
22 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
23 <li class="nav-item"><a class="nav-link active" aria-current="page" href="index.php">Home</a></li>
24 <li class="nav-item"><a class="nav-link active" aria-current="page" href="daftar.php">Daftar</a></li>
25 <li class="nav-item"><a class="nav-link active" aria-current="page" href="lihat.php">Lihat</a></li>
26 </ul>
27 </div>
28 </div>
29 </header>
30 <!-- konten utama -->
31 <main class="container mt-3" style="text-align: center;">
32 <p>Selamat Datang di Website Beasiswa</p>
33 <p>Silakan pilih jenis beasiswa untuk melanjutkan.</p>
34 <div class="row">
35 <div class="col-sm-6 mb-3 mb-sm-0">
36 <div class="card" style="margin:auto;">
37 <div class="card-body">
38 <h5 class="card-title">Beasiswa Akademik</h5>
39 <p class="card-text">persyaratan:</p>
40 <ul class="list-unstyled">
41 <li>IPK minimal 3.0</li>
42 <li>Aktif dalam organisasi kampus</li>
43 <li>Rekomendasi dari dosen</li>
44 </ul>
45 <p></p>
46 <a href="daftar.php" class="btn btn-primary">Daftar Beasiswa</a>
47 </div>
48 </div>
49 </div>
50 <div class="col-sm-6">
51 <div class="card" style="margin:auto;">
52 <div class="card-body">
53 <h5 class="card-title">Beasiswa Non-Akademik</h5>
54 <p class="card-text">persyaratan:</p>
55 <ul class="list-unstyled">
56 <li>Pengalaman dalam kegiatan sosial</li>
57 <li>Portofolio kegiatan non-akademik</li>
58 <li>Rekomendasi dari komunitas</li>
59 </ul>
60 <p></p>
61 <a href="daftar.php" class="btn btn-primary">Daftar Beasiswa</a>
62 </div>
63 </div>
64 </div>
65 </main>
66 <footer class="bg-dark text-white py-2 text-center fixed-bottom">
67 <p>©copy; 2025 Beasiswa 3M0. All rights reserved.</p>
68 </footer>
69 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/js/bootstrap.bundle.min.js" integrity="sha384-nd0U0Gzauq11F4pM11N7KfHzjVBR8d11VGR5+R9g2F2A8Z0N5405Q" crossorigin="anonymous"></script>
70 </body>
71 </html>
```

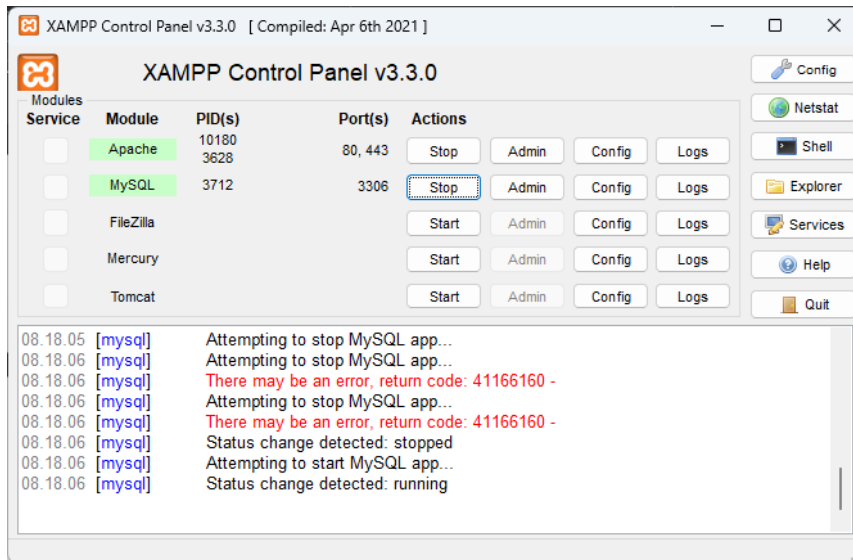
Pada file ini hanya menjelaskan tentang beasiswa pada konten utama dalam tag `<main>`, lalu juga terdapat tombol navigasi untuk berpindah halaman dalam tag `<header>` dan `<nav>`.

Logika Navigasi :

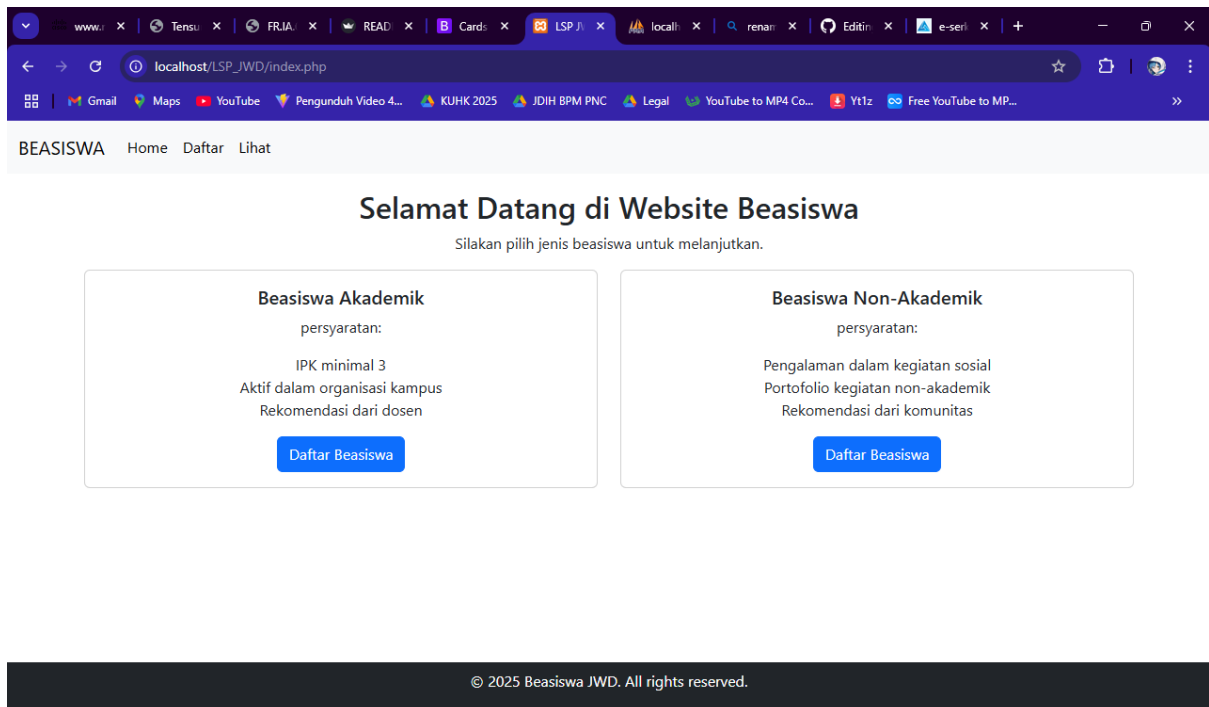
- ➔ Menu Home akan diarahkan ke file index.php
- ➔ Menu Daftar akan diarahkan ke file daftar.php
- ➔ Menu lihat akan diarahkan ke file lihat.php

Logika Konten Utama : pada konten utama terdapat 2 jenis beasiswa yang bisa didaftarkan oleh mahasiswa dengan syarat tertentu. Ketika mahasiswa klik daftar pada jenis beasiswa maka akan diarahkan ke file daftar.php.

Untuk melihat outputnya, jalankan XAMPP di Apache dan MySQL.

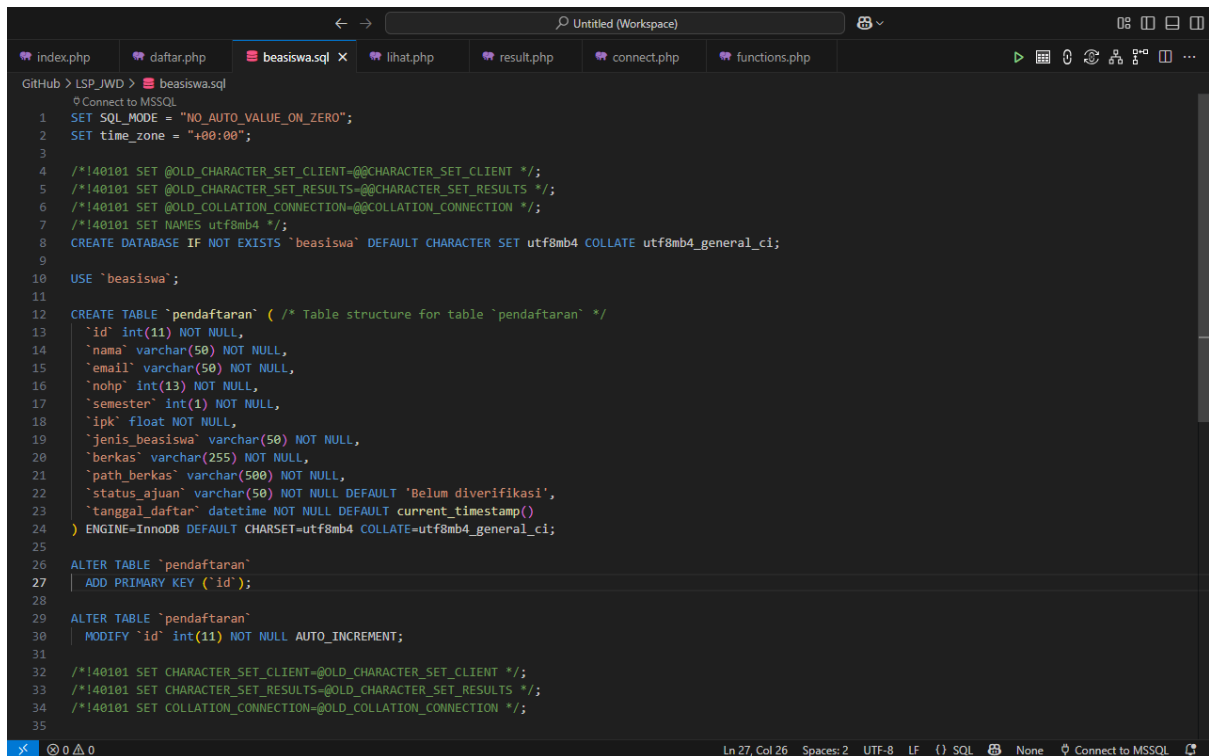


Lalu ketikkan localhost/path_file di chrome :



Membuat Database dan tabel di MySQL

Code:



```
1 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
2 SET time_zone = "+00:00";
3
4 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
5 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
6 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
7 /*!40101 SET NAMES utf8mb4 */;
8 CREATE DATABASE IF NOT EXISTS `beasiswa` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
9
10 USE `beasiswa`;
11
12 CREATE TABLE `pendaftaran` ( /* Table structure for table `pendaftaran` */
13   `id` int(11) NOT NULL,
14   `nama` varchar(50) NOT NULL,
15   `email` varchar(50) NOT NULL,
16   `nohp` int(13) NOT NULL,
17   `semester` int(1) NOT NULL,
18   `ipk` float NOT NULL,
19   `jenis_beasiswa` varchar(50) NOT NULL,
20   `berkas` varchar(255) NOT NULL,
21   `path_berkas` varchar(500) NOT NULL,
22   `status_ajuan` varchar(50) NOT NULL DEFAULT 'Belum diverifikasi',
23   `tanggal_daftar` datetime NOT NULL DEFAULT current_timestamp()
24 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
25
26 ALTER TABLE `pendaftaran`
27   ADD PRIMARY KEY (`id`);
28
29 ALTER TABLE `pendaftaran`
30   MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
31
32 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
33 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
34 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

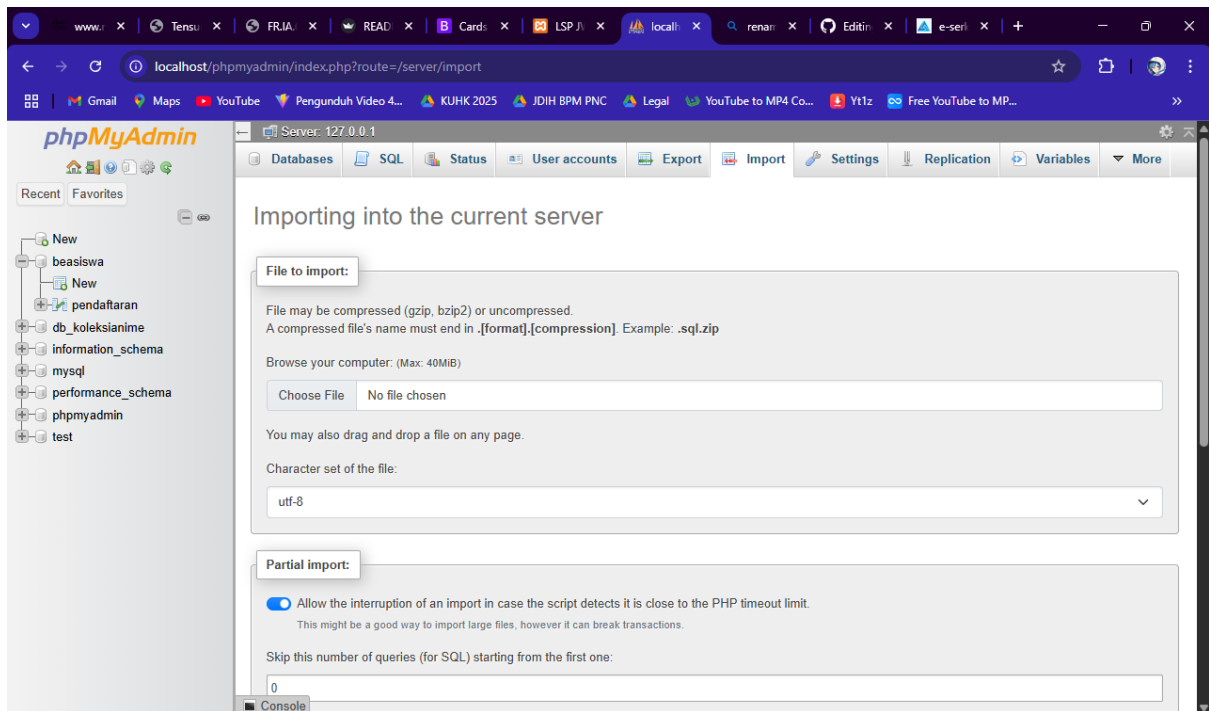
Create database akan membuat database baru jika belum ada database dengan nama beasiswa

Use database memilih database yang akan di konfigurasi

Create table akan membuat tabel dengan nama pendaftaran dengan kolom:

- Id, tipe data integer panjang 11 karakter sebagai primary key dan auto increment
- Nama, tipe data varchar panjang 50 karakter (kenapa pakai varchar? Karena ada orang yang memiliki nama dengan '. Seperti Ma'ruf Amin meskipun sekarang sudah jarang.)
- Email, tipe data varchar panjang 50 karakter
- Nohp, tipe data integer panjang 13 karakter
- Semester, tipe data integer panjang 1 karakter karena hanya valid antara 1-8
- Ipk, tipe data float karena bisa berupa bilangan pecahan
- Jenis beasiswa, tipe data varchar panjang 50 karakter
- Berkas, tipe data varchar panjang 255 karakter untuk menyimpan nama file
- Path_berkas, tipe data varchar panjang 500 karakter untuk menyimpan lokasi file
- Status_ajuan, tipe data varchar panjang 50 karakter dengan default teks "belum diverifikasi"
- Tanggal_daftar, tipe data datetime dengan default waktu data disimpan. Digunakan untuk menyimpan waktu data disimpan.

Impor kode ke localhost/phpmyadmin lalu klik import



Membuat file connect.php

```

1  <?php
2  // connect.php
3
4  $host = 'localhost'; // Host database kamu, biasanya localhost
5  $dbname = 'beasiswa'; // Nama database yang akan kamu gunakan
6  $username = 'root'; // Username database kamu
7  $password = ''; // Password database kamu (kosong jika tidak ada)
8
9  try {
10     // Membuat koneksi PDO (PHP Data Objects)
11     // PDO ini lebih modern dan aman dibanding mysql_connect() atau mysqli_connect() biasa
12     $conn = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);
13
14     // Set mode error PDO ke exception agar mudah menangani error
15     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
16
17     // echo "Koneksi database berhasil!"; // Ini bisa di-komen setelah memastikan koneksi berhasil
18 } catch (PDOException $e) {
19     // Jika koneksi gagal, tampilkan pesan error
20     die("Koneksi database gagal: " . $e->getMessage());
21 }
22 ?>

```

\$host = 'localhost'; Ini nunjukkin lokasi server database-nya.

\$dbname = 'beasiswa'; ini nama database yang mau dipakai.

\$username = 'root'; Ini nama pengguna buat ngakses database-nya. root itu username bawaan kalau belum diatur.

\$password = ''; Ini password buat username di atas. Kalo ga ada passwordnya kosongin saja.

try { ... } catch (PDOException \$e) { ... } Ini namanya blok try-catch. Fungsinya buat nangani error.

- **try:** Kode di dalam blok ini dicoba dijalankan. Kalau ada masalah, dia langsung lompat ke blok catch.
- **catch (PDOException \$e):** Kalau ada error pas nyambungin database (errornya tipe PDOException), kode di sini yang bakal jalan. Dia akan nampilin pesan "Koneksi database gagal:" ditambah detail errornya.

\$conn = new PDO("mysql:host=\$host;dbname=\$dbname;charset=utf8", \$username, \$password); Ini baris inti buat bikin koneksi ke database. Dia pake PDO (PHP Data Objects). PDO ini lebih modern, fleksibel, dan yang paling penting, lebih aman daripada mysql_connect() karena bisa mencegah serangan SQL injection.

- **"mysql:host=\$host;dbname=\$dbname;charset=utf8":** Ini string koneksinya.
- **mysql:** Tipe database-nya MySQL.
- **host=\$host:** Ngasih tau lokasi host.
- **dbname=\$dbname:** Ngasih tau nama database yang mau dihubungkan.
- **charset=utf8:** Penting buat mastiin karakter-karakter spesial kayak huruf Indonesia bisa ditangani dengan benar.

\$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); Ini untuk mengatur bagaimana PDO menangani error. Dengan PDO::ERRMODE_EXCEPTION, kalau ada error di operasi database selanjutnya, PDO akan nglempar exception, jadi kita bisa nangani errornya dengan rapi pake blok try-catch lagi.

Jadi, fingsi dari file ini adalah untuk menghubungkan file ke database mysql.

Membuat file functions.php

Code:

Fungsi **simpanPendaftaran(\$conn, \$data, \$file_info)**

```
1 <?php
2 // functions.php
3
4 // Fungsi untuk menyimpan data pendaftaran ke database
5 function simpanPendaftaran($conn, $data, $file_info) { // $conn adalah koneksi PDO, $data adalah array berisi data pendaftaran, dan $file_info adalah informasi file yang diupload
6     $nama = $data['nama'];
7     $email = $data['email'];
8     $nohp = $data['nohp'];
9     $semester = $data['semester'];
10    $ipk = $data['ipk'];
11    $jenis_beasiswa = $data['jenis_beasiswa'];
12    $nama_berkas = $file_info['berkas']; // Nama berkas yang diupload
13    $path_berkas = $file_info['path_berkas'];
14    $status_ajuan = 'Belum diverifikasi'; // Default status
15
16    try {
17        // Siapkan query untuk menyimpan data pendaftaran
18        // Menggunakan prepared statement untuk menghindari SQL Injection
19        $stmt = $conn->prepare("INSERT INTO pendaftaran (nama, email, nohp, semester, ipk, jenis_beasiswa, berkas, path_berkas, status_ajuan) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)");
20        $stmt->execute($nama, $email, $nohp, $semester, $ipk, $jenis_beasiswa, $nama_berkas, $path_berkas, $status_ajuan); // Eksekusi query dengan data yang sudah disiapkan
21        return $conn->lastInsertId(); // Mengembalikan ID pendaftaran yang baru
22    } catch (PDOException $e) {
23        throw new Exception("Gagal menyimpan data ke database: " . $e->getMessage());
24    }
25 }
```

simpanPendaftaran(\$conn, \$data, \$file_info): Fungsi ini bertanggung jawab buat menyimpan semua data pendaftaran beasiswa (nama, email, nomor HP, semester, IPK, jenis beasiswa, nama berkas, path berkas, dan status) ke dalam database.

Cara Kerjanya:

- Dia menerima tiga parameter: \$conn (koneksi ke database PDO), \$data (array yang berisi semua informasi pendaftaran dari formulir), dan \$file_info (informasi tentang berkas yang diunggah, seperti nama berkas dan lokasi penyimpanannya).
- Mengambil setiap bagian data dari array \$data dan \$file_info ke variabel terpisah biar gampang diakses.
- Status awal pendaftaran diatur sebagai 'Belum diverifikasi'.
- Menggunakan try-catch blok untuk menangani error saat menyimpan data.
- Mempersiapkan query SQL INSERT menggunakan prepared statement (\$stmt = \$conn->prepare(...)). Ini penting banget buat keamanan, karena bisa mencegah serangan SQL Injection.
- Mengeksekusi query dengan data yang sudah disiapkan (\$stmt->execute([...])).
- Setelah berhasil menyimpan, dia akan mengembalikan ID dari data pendaftaran yang baru saja tersimpan (\$conn->lastInsertId()).
- Kalau ada error saat menyimpan, dia akan melempar Exception dengan pesan error yang jelas.

Fungsi handleFileUpload(\$file, \$upload_dir)

```
27 // fungsi untuk mengelola upload file
28 function handleFileUpload($file, $upload_dir) {
29     $nama_berkas = null; // Inisialisasi variabel untuk menyimpan nama berkas
30     $path_berkas = null; // Inisialisasi variabel untuk menyimpan path berkas
31     $message = '';
32
33     if (isset($file) && $file['error'] == UPLOAD_ERR_OK) { // Cek apakah file berhasil diupload
34         $file_tmp_name = $file['tmp_name']; // Ambil nama file sementara
35         // Validasi ekstensi file
36         $allowed_extensions = ['pdf', 'docx', 'jpg', 'png']; // Daftar ekstensi yang diizinkan
37         $file_extension = pathinfo($file['name'], PATHINFO_EXTENSION); // Ambil ekstensi file
38         if (!in_array($file_extension, $allowed_extensions)) { // Cek apakah ekstensi file diizinkan
39             // Jika tidak diizinkan, kembalikan pesan error
40             return ['berkas' => null, 'path_berkas' => null, 'message' => '<div class="alert alert-danger mt-3" role="alert">Ekstensi file tidak diizinkan. Hanya file dengan ekstensi ' .
41                 implode(' ', $allowed_extensions) . ' yang diperbolehkan.</div>'];
42         }
43         $file_name = basename($file['name']); // Ambil nama file asli
44         $file_path = $upload_dir . $file_name; // Tentukan path lengkap untuk menyimpan file
45
46         if (move_uploaded_file($file['tmp_name'], $file_path)) { // Pindahkan file dari direktori sementara ke direktori upload
47             $nama_berkas = $file_name; // Simpan nama berkas
48             $path_berkas = $file_path; // Simpan path berkas
49             // Kembalikan pesan sukses
50             $message = '<div class="alert alert-success mt-3" role="alert">Berkas berhasil diupload: ' . htmlspecialchars($file_name) . '</div>';
51         } else {
52             $message = '<div class="alert alert-danger mt-3" role="alert">Gagal mengupload berkas.</div>';
53         }
54     } else if (isset($file) && $file['error'] != UPLOAD_ERR_NO_FILE) { // Jika ada error saat upload
55         // Kembalikan pesan error sesuai dengan kode error upload
56         $message = '<div class="alert alert-danger mt-3" role="alert">Error upload berkas: ' . $file['error'] . '</div>';
57     }
58     return ['berkas' => $nama_berkas, 'path_berkas' => $path_berkas, 'message' => $message];
59 }
```

handleFileUpload(\$file, \$upload_dir): Fungsi ini khusus buat ngurusin proses unggah berkas (dokumen) yang dilampirkan sama pendaftar.

Cara Kerjanya:

- Menerima \$file (informasi file yang diupload dari form HTML) dan \$upload_dir (direktori atau folder tempat file akan disimpan).
- Melakukan validasi awal apakah file berhasil diunggah (\$file['error'] == UPLOAD_ERR_OK).
- Mengecek ekstensi file yang diunggah. Dia cuma ngizinin pdf, docx, jpg, sama png. Kalo ekstensi file-nya gak diizinkan, dia bakal ngasih pesan error.
- Kalau ekstensi file diizinkan, dia nentuin nama file asli dan path lengkap tempat file akan disimpan.

- Menggunakan `move_uploaded_file()` untuk memindahkan file dari lokasi sementara (setelah diunggah) ke folder tujuan yang telah ditentukan.
- Kalau berhasil dipindahkan, dia akan mengembalikan nama berkas, path lengkap berkas, dan pesan sukses.
- Kalau gagal, atau ada error lain saat proses upload, dia akan mengembalikan pesan error yang sesuai.
- Output-nya berupa array yang berisi nama berkas (berkas), path lengkap berkas (path_berkas), dan pesan (message) buat ditampilkan ke pengguna.

Fungsi `getDataPendaftaran($conn, $id = null, $search_query = null)`

```

61 // Fungsi untuk mendapatkan data pendaftaran berdasarkan ID atau semua data dengan search
62 function getDataPendaftaran($conn, $id = null, $search_query = null) {
63     $pendaftaran_list = [];
64     $message = '';
65
66     if ($id && !$search_query) { // Jika ada ID tapi tidak ada pencarian
67         try {
68             $stmt = $conn->prepare("SELECT * FROM pendaftaran WHERE id = ?");
69             $stmt->execute([$id]); // Menggunakan parameter untuk menghindari SQL Injection
70             // Ambil satu hasil karena ID unik
71             $result = $stmt->fetch(PDO::FETCH_ASSOC); // Mengambil data pendaftaran berdasarkan ID
72             if ($result) { // Jika data ditemukan
73                 $pendaftaran_list[] = $result; // Masukkan hasil ke dalam array
74             } else { // Jika tidak ditemukan, kosongkan array
75                 $message = '<div class="alert alert-warning mt-3" role="alert">Data pendaftaran tidak ditemukan.</div>';
76             }
77         } catch (PDOException $e) { // Jika terjadi error saat mengambil data
78             $message = '<div class="alert alert-danger mt-3" role="alert">Gagal mengambil data dari database: ' . $e->getMessage() . '</div>';
79         }
80     } else if ($search_query) { // Jika tidak ada ID atau ada pencarian
81         $sql = "SELECT * FROM pendaftaran";
82         $params = []; // Inisialisasi parameter untuk query
83
84         if ($search_query) {
85             $sql .= " WHERE nama LIKE ? OR email LIKE ? OR jenis_basiswa LIKE ?";
86             $params = ["%$search_query%", "%$search_query%", "%$search_query%"]; // Mencari di kolom nama, email, dan jenis_basiswa
87         }
88         $sql .= " ORDER BY tanggal_daftar DESC"; // Urutkan dari terbaru
89
90         try {
91             $stmt = $conn->prepare($sql); // Siapkan query
92             // Eksekusi query dengan parameter pencarian jika ada
93             $stmt->execute($params);
94             $pendaftaran_list = $stmt->fetchAll(PDO::FETCH_ASSOC); // Ambil semua hasil
95             if (empty($pendaftaran_list) && empty($search_query)) { // Jika tidak ada hasil pencarian
96                 $message = '<div class="alert alert-info mt-3" role="alert">Tidak ditemukan data pendaftaran dengan kata kunci ' . htmlspecialchars($search_query) . '</div>';
97             } else if (empty($pendaftaran_list)) { // Jika tidak ada data sama sekali
98                 $message = '<div class="alert alert-info mt-3" role="alert">Belum ada data pendaftaran.</div>';
99             }
100         } catch (PDOException $e) { // Jika terjadi error saat mengambil data
101             $message = '<div class="alert alert-danger mt-3" role="alert">Gagal mengambil data dari database: ' . $e->getMessage() . '</div>';
102         }
103     }
104     return ['list' => $pendaftaran_list, 'message' => $message]; // Kembalikan array berisi daftar pendaftaran dan pesan
105 }
106 }

```

`getDataPendaftaran($conn, $id = null, $search_query = null)`: Fungsi ini buat ngambil data pendaftaran dari database. Bisa ngambil satu data spesifik berdasarkan ID, atau ngambil semua data, bahkan bisa juga nyari data berdasarkan kata kunci.

Cara Kerjanya:

- Menerima `$conn` (koneksi database), `$id` (opsional, ID pendaftaran yang mau diambil), dan `$search_query` (opsional, kata kunci pencarian).
- Kalau ada `$id` (dan gak ada `$search_query`): Dia akan ngambil satu data pendaftaran spesifik berdasarkan ID tersebut. Menggunakan prepared statement untuk keamanan.
- Kalau gak ada `$id` atau ada `$search_query`:
 - Dia akan ngambil semua data pendaftaran.
 - Kalau ada `$search_query`, dia akan menambahkan kondisi WHERE buat nyari data di kolom nama, email, atau jenis_basiswa yang mirip dengan kata kunci pencarian.
 - Hasilnya diurutkan berdasarkan tanggal_daftar secara DESC (terbaru dulu).

- Menggunakan try-catch blok untuk menangani error saat mengambil data dari database.
- Mengembalikan array yang berisi daftar pendaftaran (list) dan pesan (message) yang bisa ngasih tau status (misalnya, data gak ditemukan, atau belum ada data).

Membuat file daftar.php

Code:

```

1 <?php
2 // daftar.php
3 // file ini digunakan untuk mendaftar beasiswa
4 require 'connect.php'; // koneksi ke database
5 require 'functions.php'; // Panggil file fungsi
6
7 const IPK_MAHASISWA = 2.9; // Contoh IPK mahasiswa, bisa diambil dari database atau inputan lain
8
9 $ipk = IPK_MAHASISWA;
10 $disabled = ($ipk < 3) ? 'disabled' : ''; // Jika IPK kurang dari 3, disable inputan jenis beasiswa dan upload berkas
11
12 // Pengaman upload file
13 $upload_dir = 'uploads/'; // Direktori untuk menyimpan file yang diupload
14 if (!is_dir($upload_dir)) { //
15     mkdir($upload_dir); // Buat direktori jika belum ada
16 }
17
18 $message = ''; // Menggunakan satu variabel pesan saja
19
20 if ($_SERVER['REQUEST_METHOD'] == 'POST') { // Jika form di submit
21     if ($ipk >= 3) { //
22         $nama = htmlspecialchars($_POST['nama']);
23         $email = htmlspecialchars($_POST['email']);
24         $nohp = htmlspecialchars($_POST['nohp']); // Pastikan ini tidak dihardcode
25         $semester = htmlspecialchars($_POST['semester']);
26         $jenis_beasiswa = isset($_POST['jenis_beasiswa']) ? htmlspecialchars($_POST['jenis_beasiswa']) : '';
27
28         // Panggil fungsi handleFileUpload
29         $file_upload_result = handleFileUpload($_FILES['upload'], $upload_dir);
30         $berkas = $file_upload_result['berkas'];
31         $path_berkas = $file_upload_result['path_berkas'];
32         $message = $file_upload_result['message']; // Ambil pesan dari fungsi upload
33
34         // HANYA LANJUTKAN JIKA TIDAK ADA PESAN ERROR DARI UPLOAD FILE
35         // Periksa apakah pesan dari upload adalah pesan error (bukan sukses, atau kosong)
36         if (strpos($message, 'alert-danger') !== false) { // Cek apakah pesan berisi 'alert-danger'
37             try {
38                 // Simpan data untuk fungsi simpanPendaftaran
39                 $pendaftaran_data = [
40                     'nama' => $nama,
41                     'email' => $email,
42                     'nohp' => $nohp,
43                     'semester' => $semester,
44                     'ipk' => $ipk,
45                     'jenis_beasiswa' => $jenis_beasiswa
46                 ];
47                 $file_data = [
48                     'berkas' => $berkas,
49                     'path_berkas' => $path_berkas
50                 ];
51
52                 // Panggil fungsi simpanPendaftaran
53                 $last_id = simpanPendaftaran($conn, $pendaftaran_data, $file_data);
54
55                 // Setelah berhasil menyimpan, redirect ke halaman result.php
56                 header("location: result.php?id=" . $last_id);
57                 exit();
58             } catch (Exception $e) { // Tangkap exception dari fungsi simpanPendaftaran
59                 $message = "<div class='alert alert-danger mt-3' role='alert'>". $e->getMessage() . "</div>";
60             }
61         } else { // Jika IPK kurang dari 3, tampilkan pesan
62             $message = "<div class='alert alert-warning mt-3' role='alert'>Maaf, IPK Anda di bawah 3.0. Anda tidak dapat mendaftar beasiswa.</div>";
63         }
64     }
65 }

```

Bagian 1 php: Bagian PHP ini isinya semua logika dan proses di balik layar saat user mengakses atau mengisi formulir pendaftaran beasiswa.

require 'connect.php'; Ini buat nyambungin ke database. Jadi, semua fungsi yang butuh koneksi ke database bisa jalan.

require 'functions.php'; Ini buat manggil semua fungsi yang udah kita bikin sebelumnya di functions.php, kayak simpanPendaftaran() dan handleFileUpload().

const IPK_MAHASISWA = 3; Ini adalah nilai IPK yang di-hardcode sementara, dalam kasus ini nilainya 3. Saya buat statik untuk pengetesan saja.

\$ipk = IPK_MAHASISWA; Variabel \$ipk ini menyimpan nilai IPK yang akan dicek.

\$disabled = (\$ipk < 3) ? 'disabled' : ''; Ini logika pentingnya: Kalau IPK mahasiswa kurang dari 3, variabel \$disabled akan berisi string 'disabled'. String ini nanti dipake di bagian HTML buat nonaktifin pilihan jenis beasiswa dan field unggah berkas.

Pengaturan Direktori Unggah (\$upload_dir)

Variabel \$upload_dir diset ke 'uploads/', artinya file yang diupload bakal disimpan di folder uploads. Lalu ada pengecekan if (!is_dir(\$upload_dir)) untuk memastikan direktori itu ada. Kalau belum ada, dia akan bikin otomatis pake mkdir(\$upload_dir).

Penanganan Form Submission (if (\$_SERVER['REQUEST_METHOD'] == 'POST'))
Kode di dalam blok ini bakal jalan kalau formulir pendaftaran disubmit (dikirim). Setelah itu akan dicek IPK: Pertama, dia ngecek lagi IPK. Kalau IPK >= 3: Semua data dari formulir (nama, email, no HP, semester, jenis beasiswa) diambil dan diamankan pake htmlspecialchars() buat nyegah XSS (Cross-Site Scripting). **Fungsi handleFileUpload(\$_FILES['upload'], \$upload_dir)** dipanggil untuk memproses berkas yang diupload. Hasilnya (nama berkas, path, dan pesan) disimpan di \$file_upload_result. Kalau nggak ada pesan error dari proses upload file (strpos(\$message, 'alert-danger') === false) dia akan lanjut. Ini mencegah data disimpan kalau upload file-nya gagal.

Kalau upload berhasil (atau tidak ada error): Data pendaftaran disiapkan dalam bentuk array. **Fungsi simpanPendaftaran(\$conn, \$pendaftaran_data, \$file_data)** dipanggil buat menyimpan data ke database. Setelah berhasil disimpan, user akan di-redirect ke halaman result.php dengan membawa ID pendaftaran yang baru (**header("Location: result.php?id=" . \$last_id);**). Ini memastikan halaman tidak di-reload dan data ganda tidak tersimpan.

Kalau ada error saat menyimpan ke database (misalnya dari fungsi simpanPendaftaran), pesan errornya bakal ditangkap dan ditampilkan. Jika IPK < 3: Kalau IPK-nya di bawah 3, dia bakal nampilin pesan peringatan kalau pendaftar gak bisa ngelanjutin proses beasiswa.

\$message: Variabel ini dipake buat nampung semua pesan (sukses atau error) yang nantinya bakal ditampilkan ke user di bagian bawah formulir.

```

66 >
67 <html>
68 <head>
69 <title>ISP 360</title>
70 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-LN47Dv3652u38KpD/tr180CKT6y833zpbDp4KdLUusP697FHR/5mcr" crossorigin="anonymous">
71 <style>
72 /* Judul monastikan footer tetap di bawah */
73 body {
74     display: flex;
75     flex-direction: column;
76     min-height: 100vh;
77 }
78 main {
79     flex: 1;
80 }
81 .fixed-bottom {
82     position: fixed;
83     bottom: 0;
84     width: 100%;
85 }
86 </style>
87 </head>
88 <body>
89 <header>
90 <nav class="navbar navbar-expand-lg bg-body-tertiary">
91 <div class="container-fluid">
92 <a class="navbar-brand" href="index.php#BEASISWA"></a>
93 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
94 <span class="navbar-toggler-icon"></span>
95 </button>
96 <div class="collapse navbar-collapse" id="navbarSupportedContent">
97 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
98 <li class="nav-item"><a class="nav-link active" aria-current="page" href="index.php#Home">Home</a></li>
99 <li class="nav-item"><a class="nav-link active" aria-current="page" href="daftar.php#Daftar">Daftar</a></li>
100 <li class="nav-item"><a class="nav-link active" aria-current="page" href="lihat.php#Lihat">Lihat</a></li>
101 </ul>
102 </div>
103 </div>
104 </nav>
105 </header>
106 <main class="container mt-3">
107 <legend style="text-align: center;">Daftar Beasiswa/Legend</legend>
108 <fieldset style="width: 70%; margin: 0 auto; padding: 20px; border: 1px solid #ccc;">
109 <form action="daftar.php" method="post" enctype="multipart/form-data">
110 <table align="center" width="100%" border="0">
111 <tr>
112 <td valign="middle" class="pb-2">Masukan Nama</td>
113 <td valign="middle" class="pb-2"></td>
114 <td class="pb-2"><input class="form-control" type="text" name="nama" required</td></tr>
115 <tr>
116 <td valign="middle" class="pb-2">Masukan Email</td>
117 <td valign="middle" class="pb-2"></td>
118 <td class="pb-2"><input class="form-control" type="email" name="email" required</td></tr>
119 <tr>
120 <td valign="middle" class="pb-2">Nomor HP</td>
121 <td valign="middle" class="pb-2"></td>
122 <td class="pb-2"><input class="form-control" type="number" name="nohp" min="0" max="999999999999" required</td></tr>
123 <tr>
124 <td valign="middle" class="pb-2">Semester saat ini</td>
125 <td valign="middle" class="pb-2"></td>
126 <td class="pb-2">
127 <select class="form-select" name="semester" style="width: 150px;" required>
128 <?php for ($s = 1; $s <= 8; $s++) {>
129 <option value="<?php echo $s;"><?php echo $s;"></option>
130 <?php }>
131 </select>
132 </td>
133 </tr>
134 <tr>
135 <td valign="middle" class="pb-2">IPK terakhir</td>
136 <td valign="middle" class="pb-2"></td>
137 <td class="pb-2"><input class="form-control" type="number" name="ipk" value="<?php echo $ipk;"> readonly</td></tr>
138 <tr>
139 <td valign="middle" class="pb-2">Pilih Beasiswa</td>
140 <td valign="middle" class="pb-2"></td>
141 <td class="pb-2">
142 <select class="form-select" name="jenis_basiswa" style="width: 150px;" <?php echo $disabled;"> required>
143 <option value="akademik">Akademik</option>
144 <option value="non-akademik">Non-Akademik</option>
145 </select>
146 </td>
147 </tr>
148 <tr>
149 <td valign="middle" class="pb-2">Upload Berkas Syarat</td>
150 <td valign="middle" class="pb-2"></td>
151 <td class="pb-2"><input class="form-control" id="form-file" type="file" name="upload" <?php echo $disabled;"> required</td></tr>
152 </table>
153 <div class="text-center mt-3">
154 <input class="btn btn-lg btn-primary" style="margin-right: 50px;" type="submit" name="submit" value="Daftar" <?php echo $disabled;">
155 <input class="btn btn-lg btn-secondary" type="button" name="batal" onclick="window.location.href='index.php';" value="Batal"></div>
156 </form>
157 </fieldset>
158 <?php echo $message;">
159 </main>
160 <footer class="bg-dark text-white py-2 text-center fixed-bottom">
161 <p>©Copy, 2025 Beasiswa 360. All rights reserved.</p>
162 </footer>
163 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/js/bootstrap.bundle.min.js" integrity="sha384-nldQn0zau6pJz1f46mPLMNTKfzAVBvRd1LVR6w5R9g2fzAZDQW540SQ" crossorigin="anonymous"></script>
164 <script>
165 // Logika untuk memindahkan kursor jika IPK > 3
166 document.addEventListener('DOMContentLoaded', function() {
167     const ipkInput = document.querySelector('input[name="ipk"]');
168     const jenisBeasiswaSelect = document.querySelector('select[name="jenis_basiswa"]');
169     if (parseFloat(ipkInput.value) >= 3) { //
170         jenisBeasiswaSelect.focus();
171     }
172 });
173 </script>
174 </body>
175 </html>

```

Bagian 2 html: Halaman ini berfungsi sebagai formulir pendaftaran beasiswa yang didesain menggunakan Bootstrap untuk tampilan yang modern dan responsif.

Struktur Dasar: Terdiri dari bagian head (untuk CSS Bootstrap dan metadata) dan body (untuk konten utama). Lalu di bagian atas ada bilah navigasi sederhana dengan tautan ke "Home", "Daftar", dan "Lihat", memudahkan pengguna berpindah halaman.

Konten Utama (Formulir Pendaftaran):

- Judul di tengah: "Daftar Beasiswa".
- Sebuah fieldset membungkus formulir untuk pengelompokan visual.

- Formulir (<form>): Ini adalah inti halaman, yang akan mengirimkan data pendaftaran ke server.
- method="post": Mengirim data secara aman.
- enctype="multipart/form-data": Penting untuk mengizinkan unggahan file.
- action="daftar.php": Data akan diproses oleh file ini sendiri.
- Field Input: Berbagai kolom untuk diisi pendaftar:
 - Nama (teks)
 - Email (email)
 - Nomor HP (angka)
 - Semester saat ini (dropdown dari 1-8)
 - IPK terakhir (angka, nilainya otomatis terisi dan read-only)
 - Pilih Beasiswa (dropdown "Akademik" atau "Non-Akademik")
 - Upload Berkas Syarat (pemilihan file)
 - Kondisi IPK: Pilihan "Pilih Beasiswa" dan "Upload Berkas Syarat" bisa dinonaktifkan (disabled) secara otomatis jika nilai IPK kurang dari 3.
- Tombol: Ada tombol "Daftar" (untuk submit) dan "Batal" (untuk kembali ke halaman Home).

Pesan: Di bawah formulir, ada area untuk menampilkan pesan (sukses atau error) dari proses pendaftaran.

Footer: Di bagian paling bawah halaman, terdapat footer dengan informasi hak cipta, yang selalu berada di bagian bawah layar.

JavaScript: Ada sedikit kode JS yang secara otomatis memindahkan kursor ke pilihan jenis beasiswa jika IPK yang tertera adalah 3 atau lebih.

Output:

BEASISWA Home Daftar Lihat

Daftar Beasiswa

Masukan Nama	:	<input type="text" value="Rizki Fajar Purnomo"/>
Masukan Email	:	<input type="text" value="abiase822@gmail.com"/>
Nomor HP	:	<input type="text" value="08214386837"/>
Semester saat ini	:	<input type="text" value="4"/>
IPK terakhir	:	<input type="text" value="3"/>
Pilih Beasiswa	:	<input type="text" value="Akademik"/>
Upload Berkas Syarat	:	<input type="text" value="Choose File upscale that image.png"/>

© 2025 Beasiswa JWD. All rights reserved.

Menampilkan hasil pendaftaran terakhir dengan result.php

code:

```
1 <?php
2 // lihat.php
3 require_once 'connect.php'; // Panggil file koneksi database
4 require_once 'functions.php'; // Panggil file fungsi
5
6 $pendaftaran = null; // Inisialisasi variabel untuk data pendaftaran
7 $message = '';
8
9 if (isset($_GET['id'])) {
10     $id_pendaftaran = $_GET['id'];
11
12     // Panggil fungsi getDataPendaftaran untuk mengambil satu data detail
13     $result = getDataPendaftaran($conn, $id_pendaftaran);
14
15     if (empty($result['list'])) {
16         $pendaftaran = $result['list'][0]; // Ambil elemen pertama dari list
17     } else {
18         $message = $result['message'];
19     }
20 } else {
21     $message = '<div class="alert alert-info mt-3" role="alert">Tidak ada ID pendaftaran yang diberikan. Silakan daftar beasiswa terlebih dahulu.</div>';
22 }
23
24 <html>
25 <head>
26 <title>Hasil Pendaftaran Beasiswa - LSP JMD</title>
27 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.7/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1lkWvWvP6UzFChkawy6V1w30Uyq55LgajEefVyAIqZrr0j40Lh1BLP1vmM3M86" crossorigin="anonymous">
28 <style>
29 body { display: flex; flex-direction: column; min-height: 100vh; }
30 main { flex: 1; }
31 fixed-bottom { position: fixed; bottom: 0; width: 100%; }
32 </style>
33 </head>
34 <body>
35 <header>
36 <nav class="navbar navbar-expand-lg bg-body-tertiary">
37 <div class="container-fluid">
38 <a class="navbar-brand" href="index.php">BEASISWA</a>
39 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
40 <span class="navbar-toggler-icon"></span>
41 </button>
42 <div class="collapse navbar-collapse" id="navbarSupportedContent">
43 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
44 <li class="nav-item"><a class="nav-link active" aria-current="page" href="index.php">Home</a></li>
45 <li class="nav-item"><a class="nav-link active" aria-current="page" href="daftar.php">Daftar</a></li>
46 <li class="nav-item"><a class="nav-link active" aria-current="page" href="lihat.php">Lihat</a></li>
47 </ul>
48 </div>
49 </div>
50 </nav>
51 </header>
```

```
52 <main class="container mt-3">
53 <h2 style="text-align: center;">Hasil Pendaftaran Beasiswa</h2>
54 <?php if ($pendaftaran) {
55 <div class="card p-4" style="width: 70%; margin: 0 auto;">
56 <table class="table table-bordered">
57 <tbody>
58 <tr>
59 <th scope="row">Nama</th>
60 <td><?php echo htmlspecialchars($pendaftaran['nama']);></td>
61 </tr>
62 <tr>
63 <th scope="row">Email</th>
64 <td><?php echo htmlspecialchars($pendaftaran['email']);></td>
65 </tr>
66 <tr>
67 <th scope="row">Nomor HP</th>
68 <td><?php echo htmlspecialchars($pendaftaran['nohp']);></td>
69 </tr>
70 <tr>
71 <th scope="row">Semester Saat Ini</th>
72 <td><?php echo htmlspecialchars($pendaftaran['semester']);></td>
73 </tr>
74 <tr>
75 <th scope="row">IPK Terakhir</th>
76 <td><?php echo htmlspecialchars($pendaftaran['ipk']);></td>
77 </tr>
78 <tr>
79 <th scope="row">Jenis Beasiswa</th>
80 <td><?php echo htmlspecialchars($pendaftaran['jenis_beasiswa']);></td>
81 </tr>
82 <tr>
83 <th scope="row">Nama Berkas</th>
84 <td>
85 <?php
86 if ($pendaftaran['berkas']) {
87 echo htmlspecialchars($pendaftaran['berkas']);
88 echo " (<a href='\" . htmlspecialchars($pendaftaran['path_berkas']) . "\" target='\"_blank\">Lihat Berkas</a>";
89 } else {
90 echo "Tidak ada berkas diunggah";
91 }
92 >>
93 </td>
94 </tr>
95 <tr>
96 <th scope="row">Status Ajuan</th>
97 <td><span class="badge bg-warning text-dark"><?php echo htmlspecialchars($pendaftaran['status_ajuan']);></span></td>
98 </tr>
99 <tr>
100 <th scope="row">Tanggal Daftar</th>
101 <td><?php echo htmlspecialchars($pendaftaran['tanggal_daftar']);></td>
102 </tr>
103 </tbody>
104 </table>
105 <div class="text-center mt-3">
106 <a href="index.php" class="btn btn-primary">Kembali ke Beranda</a>
107 <a href="daftar.php" class="btn btn-secondary">Daftar Lagi</a>
108 <a href="lihat.php" class="btn btn-primary">Lihat Semua Pendaftaran</a>
109 </div>
110 <?php } else {
111 echo $message;
112 }
113 </main>
114 <footer class="bg-dark text-white py-2 text-center fixed-bottom">
115 <p>©copy; 2025 Beasiswa JMD. All rights reserved.</p>
116 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.7/dist/js/bootstrap.bundle.min.js" integrity="sha384-nbduqng201d36b20p1340759x3600" crossorigin="anonymous"></script>
117 </body>
118 </html>
```

Bagian PHP result.php: berfungsi untuk menampilkan detail satu hasil pendaftaran beasiswa setelah pendaftar berhasil submit formulir. Tujuannya mengambil dan menyiapkan data satu

pendaftaran spesifik untuk ditampilkan. Lalu memanggil connect.php untuk koneksi database dan functions.php untuk fungsi getDataPendaftaran.

Logika Utama: Mengecek apakah ada ID pendaftaran yang dikirim melalui URL (\$_GET['id']). Jika ada ID, fungsi getDataPendaftaran(\$conn, \$id_pendaftaran) dipanggil untuk mengambil detail data tersebut. Data yang ditemukan (jika ada) disimpan ke variabel \$pendaftaran. Jika ID tidak ada atau data tidak ditemukan, variabel \$message akan berisi pesan informasi atau error yang relevan.

Bagian HTML result.php

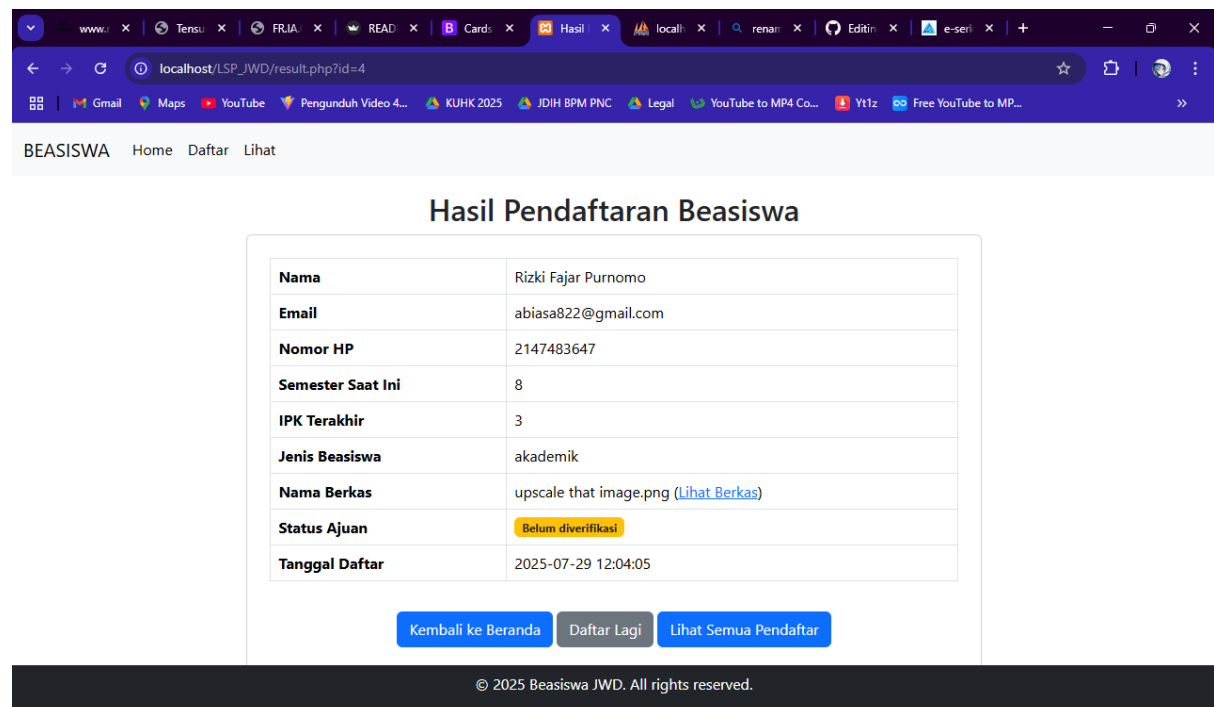
Tujuan: Menampilkan detail pendaftaran dalam format yang rapi.

Desain: Menggunakan Bootstrap untuk styling dan navbar navigasi (Home, Daftar, Lihat).

Tampilan Hasil: Jika data pendaftaran (\$pendaftaran) ditemukan, informasi detail seperti nama, email, nomor HP, semester, IPK, jenis beasiswa, nama berkas (dengan tautan "Lihat Berkas"), status ajuan, dan tanggal daftar akan ditampilkan dalam sebuah tabel di dalam kartu (.card). Status ajuan ditampilkan dengan badge berwarna (badge bg-warning). Tombol "Kembali ke Beranda", "Daftar Lagi", dan "Lihat Semua Pendaftar" disediakan di bagian bawah detail. Jika tidak ada data atau ID, \$message dari PHP akan ditampilkan (misalnya, "Tidak ada ID pendaftaran yang diberikan").

Footer: Footer tetap berada di bagian bawah halaman.

Output:



The screenshot shows a web browser window with the URL `localhost/LSP_JWD/result.php?id=4`. The page has a navigation bar with links: BEASISWA, Home, Daftar, and Lihat. The main content is titled "Hasil Pendaftaran Beasiswa" and contains a table with the following data:

Nama	Rizki Fajar Purnomo
Email	abiasa822@gmail.com
Nomor HP	2147483647
Semester Saat Ini	8
IPK Terakhir	3
Jenis Beasiswa	akademik
Nama Berkas	upscale that image.png (Lihat Berkas)
Status Ajuan	Belum diverifikasi
Tanggal Daftar	2025-07-29 12:04:05

Below the table are three buttons: "Kembali ke Beranda", "Daftar Lagi", and "Lihat Semua Pendaftar". The footer at the bottom reads: "© 2025 Beasiswa JWD. All rights reserved."

Menampilkan seluruh pendaftaran dengan lihat.php

```
1 <?php
2 // lihat.php
3 require_once 'connect.php'; // Koneksi ke database
4 require_once 'functions.php'; // Panggil file fungsi
5
6 $pendaftaran_list = []; // Inisialisasi array untuk menyimpan semua data pendaftaran
7 $search_query = ""; // Variabel untuk menyimpan kata kunci pencarian
8
9 // Ambil kata kunci pencarian jika ada
10 if (isset($_GET['search']) && !empty($_GET['search'])) {
11     $search_query = htmlspecialchars($_GET['search']);
12 }
13
14 // Panggil fungsi getDataPendaftaran
15 $result = getDataPendaftaran($conn, null, $search_query); // Memanggil tanpa ID, tapi dengan search_query
16 $pendaftaran_list = $result['list'];
17 $message = $result['message'];
18
19 // Jika ada ID di URL, ambil detail pendaftaran berdasarkan ID
20 // Logika ini dipisah agar bisa menampilkan tabel semua data atau detail satu data.
21 if (isset($_GET['id']) && !isset($_GET['search'])) {
22     $detail_result = getDataPendaftaran($conn, $_GET['id']);
23     if (!empty($detail_result['list'])) {
24         // Redirect ke result.php jika user klik detail dari tabel
25         header("location: result.php?id=" . htmlspecialchars($_GET['id']));
26         exit();
27     } else {
28         $message = $detail_result['message']; // Pesan error jika ID detail tidak ditemukan
29     }
30 }
31 //
32 <html>
33 <head>
34 <title>Daftar Beasiswa - ISIP JMD/Elites</title>
35 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Ln76d6zj6u238k9p9tR1BNCMTy83320D+pxMCLuTsP1977F4R/Sncr" crossorigin="anonymous">
36 <style>
37 body { display: flex; flex-direction: column; min-height: 100vh; }
38 main { flex: 1; }
39 <fixed-bottom { position: fixed; bottom: 0; width: 100%; }
40 .table-responsive-sm { /* untuk responsif di layar kecil */
41     overflow-x: auto;
42 }
43 </style>
44 </head>
45 <body>
46 <header>
47 <nav class="navbar navbar-expand-lg bg-body-tertiary">
48 <div class="container-fluid">
49 <a class="navbar-brand" href="index.php">BEASISWA</a>
50 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
51 <span class="navbar-toggler-icon"></span>
52 </button>
53 <div class="collapse navbar-collapse" id="navbarSupportedContent">
54 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
55 <li class="nav-item"><a class="nav-link active" aria-current="page" href="index.php">Home</a></li>
56 <li class="nav-item"><a class="nav-link active" aria-current="page" href="daftar.php">Daftar</a></li>
57 <li class="nav-item"><a class="nav-link active" aria-current="page" href="lihat.php">Lihat</a></li>
58 </ul>
59 </div>
60 </div>
61 </nav>
62 </header>
63 <main class="container mt-3">
64 <h2 style="text-align: center;">Daftar Pendaftaran Beasiswa</h2>
65
66 <div class="row mb-3 justify-content-center">
67 <div class="col-md-6">
68 <form action="lihat.php" method="get">
69 <div class="input-group">
70 <input type="text" class="form-control" placeholder="Cari Nama, Email, atau Jenis Beasiswa..." name="search" value="{?php echo htmlspecialchars($search_query); }">
71 <button class="btn btn-outline-secondary" type="submit">Cari</button>
72 <?php if (!empty($search_query)) { }>
73 <a href="lihat.php" class="btn btn-outline-danger">Reset</a>
74 <?php }>
75 </div>
76 </form>
77 </div>
78 </div>
79
80 <?php if (!empty($message)) { // Tampilkan pesan jika ada }>
81 <?php echo $message; }>
82 <?php }>
83
84 <?php if (!empty($pendaftaran_list)) { }>
85
86 <div class="table-responsive-sm">
87 <table class="table table-bordered table-striped">
88 <thead class="table-dark">
89 <tr>
90 <th>ID</th>
91 <th>Nama</th>
92 <th>Email</th>
93 <th>Nomor HP</th>
94 <th>Semester</th>
95 <th>IPK</th>
96 <th>Jenis Beasiswa</th>
97 <th>Berkas</th>
98 <th>Status Ajuan</th>
99 <th>Tanggal Daftar</th>
100 <th>Aksi</th>
101 </tr>
102 </thead>
103 <tbody>
104 <?php foreach ($pendaftaran_list as $data) { }>
105 <tr>
106 <td>{?php echo htmlspecialchars($data['id']); }</td>
107 <td>{?php echo htmlspecialchars($data['nama']); }</td>
108 <td>{?php echo htmlspecialchars($data['email']); }</td>
109 <td>{?php echo htmlspecialchars($data['nomor_hp']); }</td>
110 <td>{?php echo htmlspecialchars($data['semester']); }</td>
111 <td>{?php echo htmlspecialchars($data['ipk']); }</td>
112 <td>{?php echo htmlspecialchars($data['jenis_beasiswa']); }</td>
113 <td>
114 <?php if ($data['berkas']) { }>
115 <a href="{?php echo htmlspecialchars($data['path_berkas']); }" target="_blank" class="btn btn-sm btn-info">Lihat Berkas</a>
116 <?php } else { }>
117 Tidak Ada
118 <?php }>
119 </td>
120 <td><span class="badge bg-warning text-dark">{?php echo htmlspecialchars($data['status_ajuan']); }</span></td>
121 <td>{?php echo htmlspecialchars($data['tanggal_daftar']); }</td>
122 <td>
123 <a href="lihat.php?id={?php echo $data['id']; }" class="btn btn-sm btn-primary">Detail</a>
124 </td>
125 </tr>
126 </tbody>
127 </table>
128 </div>
129 </div>
130 <?php }>
131 </main>
132 <footer class="bg-dark text-white py-2 text-center fixed-bottom">
133 <p>KCOPY; 2025 Beasiswa JMD. All rights reserved.</p>
134 </footer>
135 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/js/bootstrap.bundle.min.js" integrity="sha384-nDuq0gZauq111WpMLhntAC124AV889911Vd0e5+R9g2fzA8ZQ0S40Q5" crossorigin="anonymous"></script>
136 </body>
137 </html>
```

lihat.php berfungsi untuk menampilkan daftar semua pendaftar beasiswa dan juga memungkinkan pencarian data pendaftaran.

Bagian PHP lihat.php

Tujuan: Mengambil daftar semua pendaftar (opsional dengan pencarian) atau me-redirect ke halaman detail jika ID spesifik diminta. Memanggil connect.php dan functions.php untuk koneksi ke database dan memanggil fungsi.

Logika Pencarian:

Mengecek apakah ada parameter search di URL. Jika ada, kata kunci pencarian disimpan di \$search_query. Lalu memanggil fungsi getDataPendaftaran(\$conn, null, \$search_query) untuk mengambil semua data, yang akan difilter jika ada \$search_query. Hasil daftar disimpan ke \$pendaftaran_list dan pesan ke \$message. Jika ada id di URL (tapi bukan pencarian), script ini akan me-redirect pengguna ke result.php untuk menampilkan detail pendaftaran tersebut.

Bagian HTML lihat.php

Tujuan: Menampilkan daftar pendaftar dalam bentuk tabel dan menyediakan fungsi pencarian.

Desain: Menggunakan Bootstrap untuk styling dan navbar navigasi.

Fungsi Pencarian:

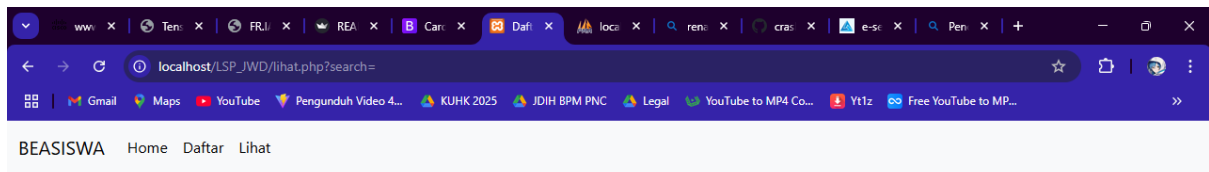
Tersedia kotak input pencarian dan tombol "Cari". Jika ada kata kunci pencarian, tombol "Reset" akan muncul untuk menghapus pencarian.

Tabel Pendaftar:

Jika \$pendaftaran_list tidak kosong, data akan ditampilkan dalam tabel responsif (table-responsive-sm). Kolom tabel meliputi: ID, Nama, Email, Nomor HP, Semester, IPK, Jenis Beasiswa, Berkas (dengan tautan "Lihat Berkas" jika ada), Status Ajuan (dengan badge), dan Tanggal Daftar. Ada kolom "Aksi" dengan tombol "Detail" yang akan mengarahkan ke result.php untuk melihat detail pendaftaran spesifik. Jika daftar kosong, \$message dari PHP akan ditampilkan (misalnya, "Belum ada data pendaftaran").

Footer: Footer tetap berada di bagian bawah halaman.

Output:

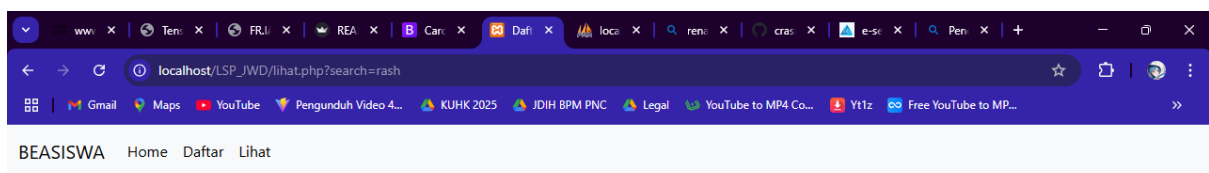


Daftar Pendaftar Beasiswa

ID	Nama	Email	Nomor HP	Semester	IPK	Jenis Beasiswa	Berkas	Status Ajuan	Tanggal Daftar	Aksi
4	Rizki Fajar Purnomo	abiasa822@gmail.com	2147483647	8	3	akademik	Lihat Berkas	Belum diverifikasi	2025-07-29 12:04:05	Detail
3	wwerty	wqrew@email.com	875465687	3	3.4	akademik	Lihat Berkas	Belum diverifikasi	2025-07-29 08:24:27	Detail
2	Rash	abcd@gmail.com	2147483647	5	3.4	non-akademik	Lihat Berkas	Belum diverifikasi	2025-07-28 22:45:28	Detail
1	Rizki Fajar Purnomo	abiasa822@gmail.com	2147483647	2	3	akademik	Lihat Berkas	Belum diverifikasi	2025-07-28 22:05:27	Detail

© 2025 Beasiswa JWD. All rights reserved.

Mencari data dengan nama rash



Daftar Pendaftar Beasiswa

ID	Nama	Email	Nomor HP	Semester	IPK	Jenis Beasiswa	Berkas	Status Ajuan	Tanggal Daftar	Aksi
2	Rash	abcd@gmail.com	2147483647	5	3.4	non-akademik	Lihat Berkas	Belum diverifikasi	2025-07-28 22:45:28	Detail

© 2025 Beasiswa JWD. All rights reserved.

Source code: https://github.com/Rashfox/LSP_JWD.git