

BAB IV

HASIL DAN PEMBAHASAN

4.1 Tahapan Penelitian

Penelitian ini mengusulkan penggunaan CAPTCHA permainan flow sebagai solusi inovatif untuk meningkatkan keamanan sistem online. CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) telah diterapkan dengan mengintegrasikan elemen permainan flow, yang mencakup berbagai tugas seperti puzzle, teka-teki, atau tugas interaktif lainnya. Tujuan utama penelitian ini adalah merancang dan mengimplementasikan CAPTCHA yang tidak hanya dapat membedakan manusia dan komputer secara efektif tetapi juga memberikan pengalaman pengguna yang positif. Metode perancangan ini bertujuan untuk menciptakan tugas-tugas yang dapat diselesaikan dengan kemampuan kognitif manusia namun sulit diotomasi oleh bot, sehingga meningkatkan tingkat keamanan dalam verifikasi manusia. Selain itu, penelitian ini juga mengevaluasi keefektifan CAPTCHA permainan flow dalam menghadapi serangan otomatis, serta menganalisis sejauh mana pengguna merasa terlibat dan nyaman dalam proses verifikasi ini. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi positif dalam pengembangan metode CAPTCHA yang lebih aman dan dapat diterima oleh pengguna.

4.1.1 Halaman Login

URL yang terlihat adalah `http://localhost/TAfreeflow/index.php`. "localhost" menunjukkan bahwa aplikasi ini sedang dijalankan pada server

lokal di komputer Anda sendiri, bukan di server eksternal atau online.

"TAfreeflow" kemungkinan adalah nama dari aplikasi atau proyek yang sedang Anda kerjakan. Aplikasi ini bisa saja berhubungan dengan tugas akhir (TA) atau proyek lain yang Anda kerjakan secara lokal.

index.php menunjukkan bahwa halaman ini ditulis menggunakan bahasa pemrograman PHP, yang umumnya digunakan untuk pengembangan web dinamis.

Formulir Login:

Formulir login ini ditempatkan di tengah halaman dengan desain yang minimalis dan bersih, memastikan fokus pengguna tetap pada formulir tersebut.

Field Username: Ini adalah input teks di mana pengguna harus memasukkan nama pengguna mereka. Label "Username" ditampilkan di atas kotak input untuk memberikan petunjuk kepada pengguna.

Field Password: Ini adalah input kata sandi di mana pengguna harus memasukkan kata sandi mereka. Untuk keamanan, karakter dalam input ini digantikan dengan bintang atau titik (●●●●●●●●).

Tombol Sign In: Tombol ini, saat diklik, akan mengirimkan data yang dimasukkan dalam field username dan password ke server untuk proses autentikasi. Tombol ini berwarna hitam dengan teks putih yang kontras, membuatnya mudah terlihat dan diakses oleh pengguna.

Tautan Registrasi:

Di bawah tombol "Sign In", terdapat teks yang berbunyi "Don't have an account? Register here".

"Register here" adalah tautan yang dapat diklik oleh pengguna yang belum memiliki akun untuk diarahkan ke halaman pendaftaran. Ini memberikan opsi kepada pengguna baru untuk membuat akun sebelum mencoba login.

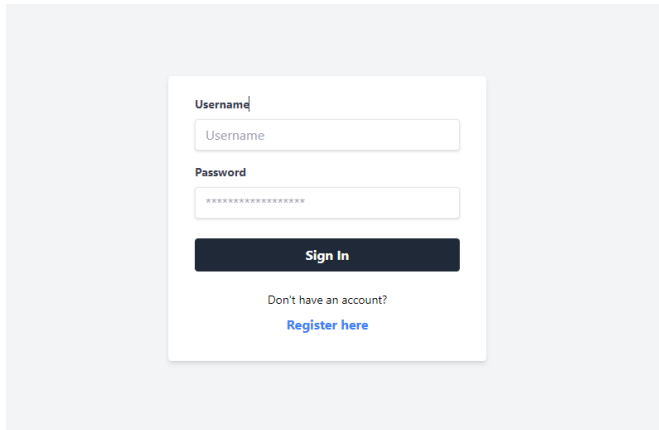
Desain dan Antarmuka:

Desain halaman ini sangat sederhana dan fungsional, dengan latar belakang putih yang luas dan elemen-elemen antarmuka yang terpusat.

Fokus pada kegunaan, memastikan bahwa pengguna baru maupun yang sudah terdaftar dapat dengan mudah memahami bagaimana cara menggunakan halaman ini.

Penggunaan font yang sederhana dan jelas, dengan kontras warna yang baik antara teks dan latar belakang, meningkatkan keterbacaan.

Halaman ini merupakan bagian penting dari sistem autentikasi, di mana pengguna diharuskan memasukkan kredensial mereka untuk mendapatkan akses ke bagian-bagian lain dari aplikasi web. Proses ini membantu memastikan bahwa hanya pengguna yang sah yang dapat mengakses informasi dan fitur yang dilindungi.



Gambar 4. 1 Menu Utama

4.1.2 Perancangan Sistem

Perancangan sistem dimulai dari pembuatan halaman login, halaman registrasi, halaman permainan, dan dashboard.

1. Halaman Login

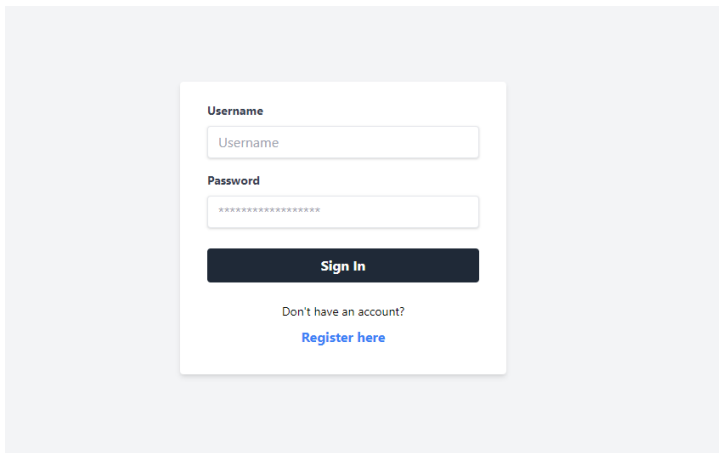
```

<DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Login | Elixir</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="main.js" defer"></script>
</head>
<body class="bg-gray-100 flex flex-center justify-center h-screen">
  <div class="bg-white shadow-md rounded px-8 pt-6 pb-8 flex flex-col w-80">
    <div class="text-center">
      <h2>Sign In</h2>
    </div>
    <div class="mb-3">
      <input type="text" id="username" value="" class="form-control" required="" />
    </div>
    <div class="mb-3">
      <input type="password" id="password" value="" class="form-control" required="" />
    </div>
    <div class="text-center">
      <button type="button" class="btn btn-primary" href="#">Sign In</button>
    </div>
    <div class="text-center">
      <p>Don't have an account?</p>
    </div>
    <div class="text-center">
      <a href="#" class="text-decoration: none; color: #007bff; font-weight: bold;">Register here</a>
    </div>
  </div>
</body>
</html>
  
```

Gambar 4. 2 Halaman Login

Kode HTML ini membangun halaman login dengan latar belakang abu-abu muda dan konten di tengah layar

menggunakan Flexbox. Di dalam kontainer utama berwarna putih dengan bayangan dan sudut membulat terdapat formulir login POST yang mencakup input tersembunyi untuk `userId`, input teks untuk `username` dan `password`, serta tombol "Sign In" yang berwarna abu-abu tua dan berubah menjadi biru saat dihover, semua dengan styling dari Tailwind CSS. Jika terdapat error, pesan error akan ditampilkan dalam teks merah di bawah tombol. Terdapat juga teks dan link yang mengarahkan pengguna ke halaman pendaftaran (`register.php`). Selain itu, JavaScript digunakan untuk mengambil `user_id` dari sesi PHP dan menyimpannya ke dalam variabel `userId`.



Gambar 4. 3 Hasil Output

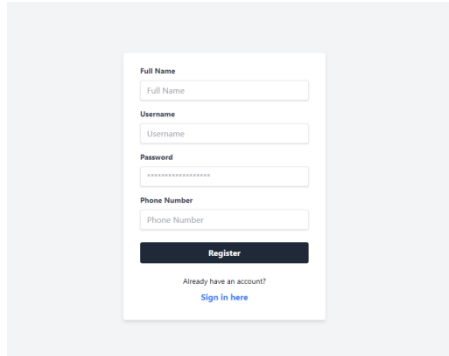
Berikut adalah hasil dari *source code* halaman login diatas.

2. Halaman Registrasi

```
<script src="/js/jquery.js"></script>
</script>
<script src="/js/main.js"></script>
</script>
</body>
</html>
```

Gambar 4. 4 Halaman Registrasi

Kode HTML ini membangun halaman pendaftaran dengan latar belakang abu-abu muda dan konten di tengah layar menggunakan Flexbox. Di dalam kontainer utama berwarna putih dengan bayangan dan sudut membulat terdapat formulir pendaftaran POST yang mencakup input untuk nama lengkap, username, password, dan nomor telepon, semua dengan styling dari Tailwind CSS. Tombol "Register" berwarna abu-abu tua dan berubah menjadi biru saat dihover. Jika terdapat error, pesan error akan ditampilkan dalam teks merah di bawah tombol. Terdapat juga teks dan link yang mengarahkan pengguna ke halaman login (index.php).



Gambar 4. 5 Hasil Output

Berikut adalah hasil *source code* halaman registrasi diatas.

3. Halaman Permainan

1. Pembuatan Garis 5x5 Permainan

1.1 Mendefinisikan *class*

```
const CSIZE = 45;
class Grid {
  constructor(rows, cols, board) {
    this.rows = rows;
    this.cols = cols;
    this.width = CSIZE * 10;
    this.height = CSIZE * 10;
    this.cellWidth = Math.floor(this.width / this.rows);
    this.cellHeight = Math.floor(this.height / this.cols);
    this.board = board;
    this.pipes = [];
  }
}
```

Gambar 4. 6 Pembuatan Grid

Pemrograman Berorientasi Objek (OOP) adalah paradigma pemrograman yang berfokus pada objek sebagai elemen dasar penyusun program. Dalam OOP, objek memiliki properti (atribut) yang

merepresentasikan datanya dan metode (fungsi) yang merepresentasikan perilakunya.

1.2 Mengakses properti objek

```
function buildGrid() {
  currentLevel = (currentLevel + 1) % levels.length; // Ping
  let level = levels[currentLevel];
  let size = level.length;
  grid = new Grid(5, 5, level);
  currentPipe = new Pipe([], EMPTY);
  startTime = millis();
}
```

Gambar 4. 7 Halaman pembuatan jumlah Grid

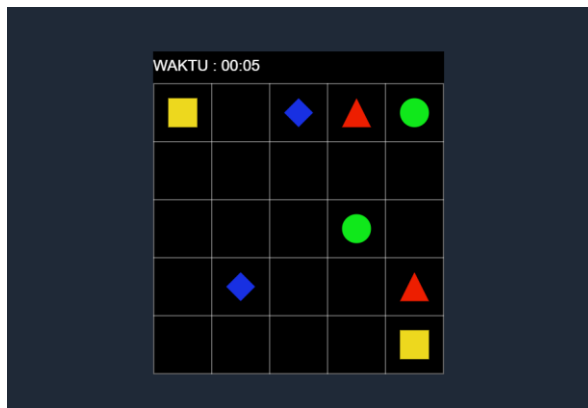
Memberikan parameter (5,5,level) parameter untuk diisi oleh *class* yang sudah di definisikan diatas

2. Pembuatan pola bentuk bidang

```
drawCircle() {
  noStroke();
  for (let row = 0; row < this.board.length; rows++) {
    for (let col = 0; col < this.board[row].length; cols++) {
      let color = this.board[row][col];
      if (color == EMPTY) {
        continue;
      }
      fillWithColor(color, (r, g, b) => {
        fill(r, g, b);
      });
      switch(color) {
        case RED:
          triangle(col * this.cellWidth + this.cellWidth / 2, row * this.cellHeight + this.cellHeight / 2 - CSIZE / 2,
            col * this.cellWidth + this.cellWidth / 2 - CSIZE / 2, row * this.cellHeight + this.cellHeight / 2 + CSIZE / 2,
            col * this.cellWidth + this.cellWidth / 2 + CSIZE / 2, row * this.cellHeight + this.cellHeight / 2 + CSIZE / 2);
          break;
        case GREEN:
          ellipse(col * this.cellWidth + this.cellWidth / 2, row * this.cellHeight + this.cellHeight / 2, CSIZE, CSIZE);
          break;
        case BLUE:
          let diamondSize = CSIZE / 2;
          quad(col * this.cellWidth + this.cellWidth / 2, row * this.cellHeight + this.cellHeight / 2 - diamondSize / 2,
            col * this.cellWidth + this.cellWidth / 2 - diamondSize / 2, row * this.cellHeight + this.cellHeight / 2,
            col * this.cellWidth + this.cellWidth / 2, row * this.cellHeight + this.cellHeight / 2 + diamondSize / 2,
            col * this.cellWidth + this.cellWidth / 2 + diamondSize / 2, row * this.cellHeight + this.cellHeight / 2);
          break;
        case YELLOW:
          rect(col * this.cellWidth + this.cellWidth / 2 - CSIZE / 2, row * this.cellHeight + this.cellHeight / 2 - CSIZE / 2, CSIZE, CSIZE);
          break;
        case ORANGE:
          quad(col * this.cellWidth + this.cellWidth * 0.25, row * this.cellHeight + this.cellHeight * 0.75,
            col * this.cellWidth + this.cellWidth * 0.75, row * this.cellHeight + this.cellHeight * 0.25,
            col * this.cellWidth + this.cellWidth * 0.9, row * this.cellHeight + this.cellHeight * 0.25,
            col * this.cellWidth + this.cellWidth * 0.1, row * this.cellHeight + this.cellHeight * 0.25);
          break;
        default:
          break;
      }
    }
  }
}
```

Gambar 4. 8 Halaman pembuatan bidang

Metode `drawCircles` ini tidak menggunakan garis tepi dan iterasi melalui setiap elemen dalam `this.board` untuk menggambar bentuk-bentuk berdasarkan nilai warna di setiap sel. Jika warna adalah `EMPTY`, iterasi akan dilanjutkan tanpa menggambar. Fungsi `fillWithColor` digunakan untuk mengatur warna pengisian berdasarkan warna yang ditemukan. Jika warna adalah `RED`, digambar segitiga pada posisi sel yang dihitung; jika `GREEN`, digambar elips; jika `BLUE`, digambar belah ketupat; jika `YELLOW`, digambar persegi; dan jika `ORANGE`, digambar belah ketupat dengan posisi dan ukuran yang berbeda. Posisi dan ukuran setiap bentuk dihitung berdasarkan indeks baris dan kolom serta ukuran sel yang ditentukan oleh `this.cellWidth` dan `this.cellHeight`, dengan ukuran bentuk dikendalikan oleh konstanta `CSIZE`.



Gambar 4. 9 Hasil Output

Berikut adalah hasil dari *source code* pembuatan bentuk bidang.

3. Pembuatan *pipe* (Garis penghubung bentuk bidang)

```
draw(grid) {
  for (let i = 0; i < this.vectors.length; i++) {
    let start = this.vectors[i];

    fillWithColor(this.color, (r, g, b) => {
      fill(r, g, b, 50);
      stroke(r, g, b);
    });

    strokeWeight(2);
    rect(start.x * grid.cellWidth, start.y * grid.cellHeight, grid.cellWidth, grid.cellHeight);

    if (i < this.vectors.length - 1) {
      let end = this.vectors[i + 1];

      fillWithColor(this.color, (r, g, b) => stroke(r, g, b));
      strokeWeight(12);
      line(start.x * grid.cellWidth + grid.cellWidth / 2,
           start.y * grid.cellHeight + grid.cellHeight / 2,
           end.x * grid.cellWidth + grid.cellWidth / 2,
           end.y * grid.cellHeight + grid.cellHeight / 2);
      noStroke();
    }
  }
}
```

Gambar 4. 10 Pembuatan *pipe*

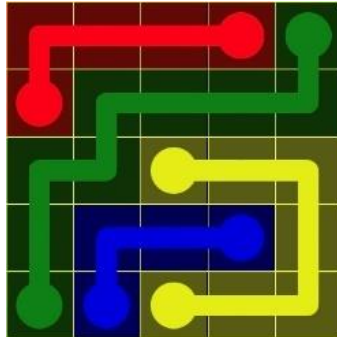
Metode `draw(grid)` ini menggambar bentuk-bentuk dan garis-garis berdasarkan array `this.vectors` dan objek `grid` yang memiliki properti `cellWidth` dan `cellHeight`. Untuk setiap vektor dalam `this.vectors`, metode ini mengisi dan menggambar persegi panjang pada posisi yang dihitung menggunakan koordinat vektor dikalikan dengan lebar dan tinggi sel `grid`. Warna pengisian dan garis ditentukan oleh `this.color` dan transparansi 50 untuk pengisian. Setelah menggambar persegi panjang, jika vektor bukan yang terakhir dalam array, metode ini menggambar garis tebal antara vektor saat ini dan vektor berikutnya menggunakan warna yang sama. Posisi tengah setiap sel digunakan sebagai titik awal dan akhir garis, memastikan garis berada di tengah-tengah setiap sel.

Biru: Garis biru dimulai dari titik biru di dekat sudut kiri atas dan berakhir pada titik biru di dekat tengah, dengan pola berbelok beberapa kali.

Merah: Garis merah dimulai dari titik merah di sisi kiri tengah dan berakhir pada titik merah di sudut kanan bawah.

Hijau: Garis hijau dimulai dari titik hijau di sisi kanan tengah dan berakhir pada titik hijau di tengah, juga dengan beberapa belokan.

2. Model 2



Gambar 4. 13 Model 2

Berikut adalah pola pada gambar tersebut:

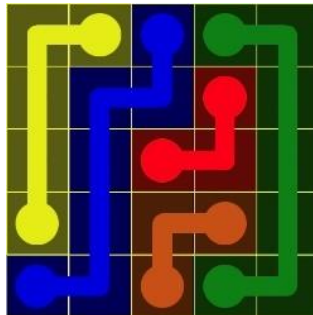
Merah: Garis merah dimulai dari titik merah di sudut kiri atas dan berakhir pada titik merah di sudut kanan atas, mengikuti jalur horizontal.

Hijau: Garis hijau dimulai dari titik hijau di sudut kiri bawah dan berakhir pada titik hijau di sudut kanan atas, melalui jalur yang berkelok-kelok.

Kuning: Garis kuning dimulai dari titik kuning di tengah kiri dan berakhir pada titik kuning di sisi kanan bawah, dengan pola berbelok beberapa kali.

Biru: Garis biru dimulai dari titik biru di sudut kiri bawah dan berakhir pada titik biru di sisi kanan tengah, dengan pola berbelok beberapa kali.

3. Model 3



Gambar 4. 14 Model 3

Berikut adalah pola pada gambar tersebut:

Kuning: Garis kuning dimulai dari titik kuning di sudut kiri atas dan berakhir pada titik kuning di sudut kiri bawah, melalui jalur yang berkelok-kelok.

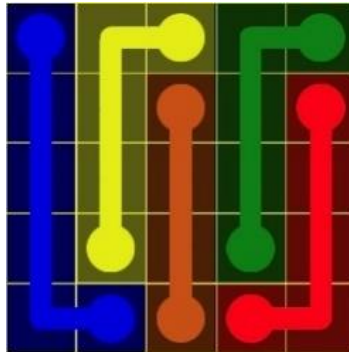
Biru: Garis biru dimulai dari titik biru di sisi tengah kiri dan berakhir pada titik biru di sisi tengah atas, dengan pola berbelok beberapa kali.

Merah: Garis merah dimulai dari titik merah di sisi tengah bawah dan berakhir pada titik merah di sisi tengah kanan, melalui jalur yang berkelok-kelok.

Hijau: Garis hijau dimulai dari titik hijau di sisi kanan tengah dan berakhir pada titik hijau di sudut kanan atas, mengikuti jalur vertikal.

Oranye: Garis oranye dimulai dari titik oranye di tengah bawah dan berakhir pada titik oranye di tengah atas, dengan pola berbelok beberapa kali.

4. Model 4



Gambar 4. 15 Model 4

Berikut adalah pola pada gambar tersebut:

Kuning: Garis kuning dimulai dari titik kuning di sudut kiri atas dan berakhir pada titik kuning di sudut kiri bawah, melalui jalur yang berkelok-kelok.

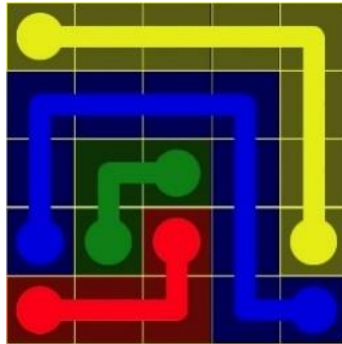
Biru: Garis biru dimulai dari titik biru di sudut kiri bawah dan berakhir pada titik biru di sudut kanan atas, melalui jalur yang berkelok-kelok.

Merah: Garis merah dimulai dari titik merah di sudut kiri atas dan berakhir pada titik merah di sudut kanan bawah, melalui jalur yang berkelok-kelok.

Hijau: Garis hijau dimulai dari titik hijau di sisi kanan atas dan berakhir pada titik hijau di sisi kanan bawah, mengikuti jalur vertikal.

Oranye: Garis oranye dimulai dari titik oranye di tengah kiri dan berakhir pada titik oranye di tengah kanan, dengan pola berbelok beberapa kali.

5. Model 5



Gambar 4. 16 Model 5

Berikut adalah pola pada gambar yang baru Anda unggah:

Kuning: Garis kuning dimulai dari titik kuning di sudut kiri atas dan berakhir pada titik kuning di sudut kanan

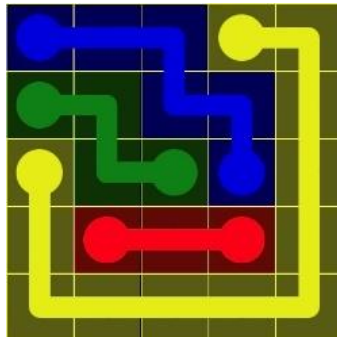
bawah. Garis ini berjalan horizontal ke kanan hingga mencapai kolom terakhir, lalu menurun ke baris terakhir.

Biru: Garis biru dimulai dari titik biru di sudut kiri bawah dan berakhir pada titik biru di sudut kanan atas. Garis ini naik secara vertikal ke baris kedua dari atas, lalu bergerak horizontal ke kanan hingga mencapai kolom terakhir

Hijau: Garis hijau dimulai dari titik hijau di baris ketiga dari bawah di kolom ketiga dari kiri dan berakhir pada titik hijau di baris ketiga dari atas di kolom ketiga dari kiri. Garis ini bergerak secara vertikal naik dari titik awal hingga mencapai titik akhir.

Merah: Garis merah dimulai dari titik merah di baris kedua dari bawah di kolom kedua dari kiri dan berakhir pada titik merah di baris kedua dari atas di kolom kedua dari kiri. Garis ini bergerak secara vertikal naik dari titik awal hingga mencapai titik akhir.

6. Model 6



Gambar 4. 17 Model 6

Berikut adalah pola pada gambar yang baru Anda unggah:

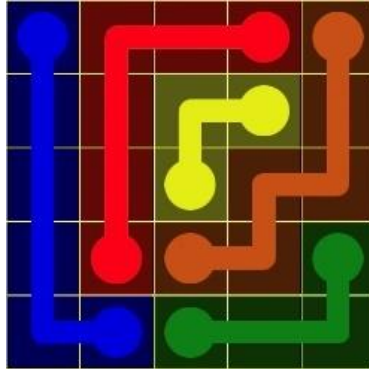
Kuning: Garis kuning dimulai dari titik kuning di sudut kiri bawah dan berakhir pada titik kuning di sudut kanan atas. Garis ini berjalan vertikal ke atas hingga mencapai baris kedua dari atas, lalu bergerak horizontal ke kanan hingga mencapai kolom terakhir, lalu naik satu baris lagi dan bergerak ke kiri hingga mencapai kolom kedua dari kiri, kemudian naik satu kotak lagi dan bergerak ke kanan hingga mencapai titik kuning.

Biru: Garis biru dimulai dari titik biru di sudut kiri atas dan berakhir pada titik biru di sudut kanan tengah. Garis ini berjalan horizontal ke kanan hingga mencapai kolom kedua dari kanan, lalu menurun ke baris ketiga dari atas, kemudian bergerak horizontal ke kiri hingga mencapai kolom kedua dari kiri, lalu menurun satu baris lagi dan bergerak ke kanan hingga mencapai titik biru.

Hijau: Garis hijau dimulai dari titik hijau di sisi kiri tengah dan berakhir pada titik hijau di sisi kanan tengah. Garis ini bergerak horizontal dari kiri ke kanan, lalu menurun satu baris, kemudian bergerak horizontal ke kiri hingga mencapai titik hijau.

Merah: Garis merah dimulai dari titik merah di sisi kiri bawah dan berakhir pada titik merah di sisi kanan bawah. Garis ini berjalan horizontal dari kiri ke kanan tanpa ada belokan.

7. Model 7



Gambar 4. 18 Model 7

Berikut adalah penjelasan pola yang ada pada gambar:

Merah: Menghubungkan dua titik merah di kiri atas dan tengah kanan dengan melewati beberapa kotak secara vertikal dan horizontal.

Biru: Menghubungkan dua titik biru di kiri bawah dan kiri atas dengan melewati beberapa kotak secara vertikal dan horizontal.

Hijau: Menghubungkan dua titik hijau di kiri bawah dan kanan bawah dengan melewati beberapa kotak secara vertikal dan horizontal.

Oranye: Menghubungkan dua titik oranye di tengah bawah dan tengah atas dengan melewati beberapa kotak secara vertikal dan horizontal.

Kuning: Menghubungkan dua titik kuning di tengah-tengah gambar dengan bentuk yang cukup kompleks dengan beberapa belokan.

5. Pembuatan Halaman Dashboard

1. Halaman Dashboard

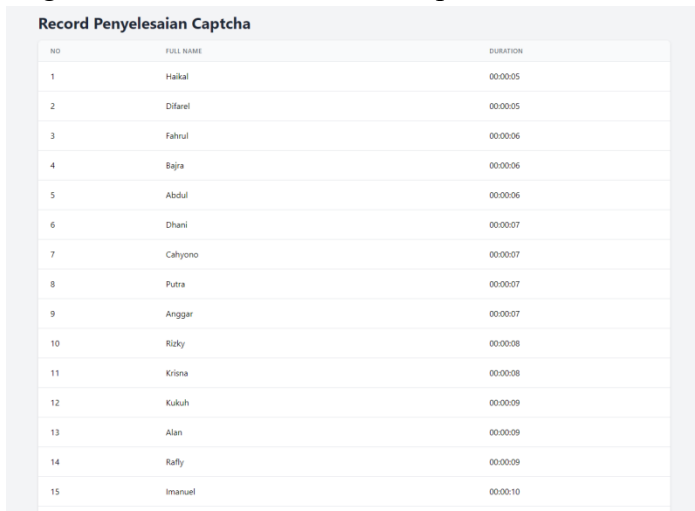
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title Dashboard </title>
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.10.0/dist/tailwind.min.css" rel="stylesheet">
</head>
<body class="bg-gray-800">
  <div class="bg-gray-800 p-6">
    <div class="max-w-7xl mx-auto">
      <div class="flex items-center justify-between">
        <div class="font-size-center font-weight-6 text-white m-0">
          <span class="font-size-bold text-xl tracking-tight">Dashboard </span>
        </div>
        <div class="block">
          <a href="logout.php" class="inline-block text-sm px-4 py-2 leading-none rounded text-white border-white hover:border-transparent hover:text-gray-400 hover:bg-white m-0">Logout </a>
        </div>
      </div>
      <div class="max-w-7xl mx-auto py-6 sm:px-6 lg:px-8">
        <div class="px-4 py-6 sm:px-0">
          <div class="text-3xl font-size-1 font-weight-6 text-gray-400 m-0">Welcome to Dashboard </div>
          <div class="bg-white shadow overflow-hidden sm:rounded-lg">
            <div class="px-4 py-3 sm:px-0">
              <div class="text-lg leading-6 font-medium text-gray-900">User Information </div>
              <div class="border-1 border-gray-200">
                <div class="bg-gray-800 p-4 sm:px-6 sm:grid sm:grid-cols-2 sm:gap-4 sm:px-0">
                  <div class="text-sm font-medium text-gray-400">Full Name </div>
                  <div class="m-1 text-sm text-gray-400 sm:col-span-2">{{php echo isset($_SESSION['nama_lengkap']) ? $_SESSION['nama_lengkap'] : '' ;}} </div>
                  <div class="text-sm font-medium text-gray-400">Email </div>
                  <div class="m-1 text-sm text-gray-400 sm:col-span-2">{{php echo isset($_SESSION['email']) ? $_SESSION['email'] : '' ;}} </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

Gambar 4. 19 Halaman Dashboard

Kode HTML ini membangun sebuah halaman dashboard dengan gaya yang responsif menggunakan Tailwind CSS. Berikut adalah penjelasan singkatnya dalam satu paragraf:

Halaman dashboard ini dimulai dengan sebuah navigasi berwarna abu-abu tua (bg-gray-800) yang berisi teks "Dashboard" dan sebuah link untuk logout. Konten utama berada di dalam div dengan kelas max-w-7xl mx-auto py-6 sm:px-6 lg:px-8 untuk mengatur lebar maksimum dan padding. Di dalamnya terdapat judul besar "Welcome to Dashboard" diikuti dengan sebuah kartu informasi berwarna putih dengan bayangan (bg-white shadow). Kartu ini menampilkan informasi pengguna yang diambil dari sesi PHP, seperti nama lengkap dan durasi penyelesaian captcha, dalam format grid tiga kolom yang responsif. Informasi ini ditampilkan dalam elemen-elemen dengan styling

sm:px-0. Tabel tersebut diatur dengan gaya Tailwind CSS dan memiliki header yang terdiri dari kolom "NO", "Full Name", dan "Duration". Dalam bagian body tabel, PHP digunakan untuk memeriksa apakah ada data yang tersedia dalam hasil query (\$result). Jika data tersedia, tabel diisi dengan baris yang mencantumkan nomor, nama lengkap, dan durasi penyelesaian Captcha dari setiap baris data yang diperoleh dari database. Jika tidak ada data, tabel menampilkan pesan "No data found" pada satu baris yang menggabungkan ketiga kolom. Tabel ini memiliki latar belakang berwarna putih dengan bayangan dan sudut membulat, serta pembagian garis-garis abu-abu untuk menandai pemisahan antar baris.



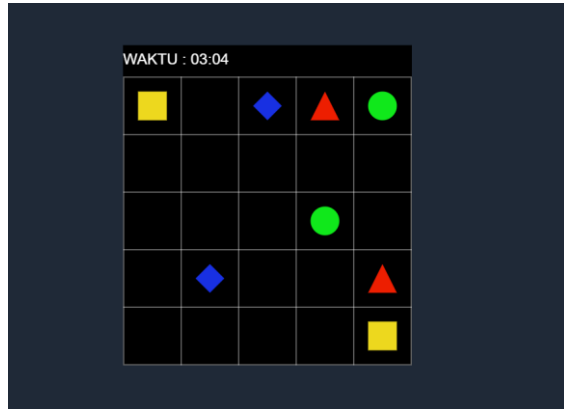
NO	FULL NAME	DURATION
1	Haikal	00:00:05
2	Difarel	00:00:05
3	Fahrul	00:00:06
4	Bajra	00:00:06
5	Abdul	00:00:06
6	Dhani	00:00:07
7	Cahyono	00:00:07
8	Putra	00:00:07
9	Anggar	00:00:07
10	Rizky	00:00:08
11	Krisna	00:00:08
12	Kukuh	00:00:09
13	Alan	00:00:09
14	Raffy	00:00:09
15	Immanuel	00:00:10

Gambar 4. 22 Hasil Output

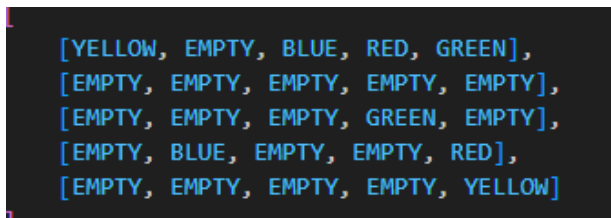
Berikut adalah hasil *source code* dari halaman *record*.

4.1.3 Permainan Flow

Uji coba pada menu utama ini merupakan akses yang akan digunakan.



Gambar 4. 23 Permainan Flow 1



Gambar 4. 24 Source Code 1

Baris 1 ([YELLOW, EMPTY, BLUE, RED, GREEN]): Baris pertama dari matriks.

- Elemen pertama (YELLOW) berada di kolom pertama.
- Elemen kedua (EMPTY) menunjukkan bahwa kolom kedua kosong.
- Elemen ketiga (BLUE) berada di kolom ketiga.

- Elemen keempat (RED) berada di kolom keempat.
- Elemen kelima (GREEN) berada di kolom kelima.

Baris 2 ([EMPTY, EMPTY, EMPTY, EMPTY, EMPTY]):
Baris kedua dari matriks.

- Seluruh elemen pada baris ini adalah EMPTY, menunjukkan bahwa semua kotak pada baris ini kosong.

Baris 3 ([EMPTY, EMPTY, EMPTY, GREEN, EMPTY]):
Baris ketiga dari matriks.

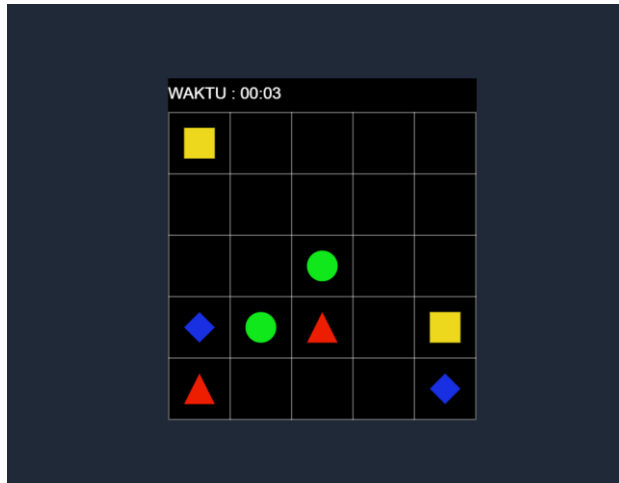
- Elemen keempat (GREEN) berada di kolom keempat, sementara yang lainnya kosong.

Baris 4 ([EMPTY, BLUE, EMPTY, EMPTY, RED]): Baris
keempat dari matriks.

- Elemen kedua (BLUE) berada di kolom kedua.
- Elemen kelima (RED) berada di kolom kelima, sementara yang lainnya kosong.

Baris 5 ([EMPTY, EMPTY, EMPTY, EMPTY, YELLOW]):
Baris kelima dari matriks.

- Elemen kelima (YELLOW) berada di kolom kelima, sementara yang lainnya kosong.



Gambar 4. 25 Permainan Flow 2

```
[YELLOW, EMPTY, EMPTY, EMPTY, EMPTY],
[EMPTY, EMPTY, EMPTY, EMPTY, EMPTY],
[EMPTY, EMPTY, GREEN, EMPTY, EMPTY],
[BLUE, GREEN, RED, EMPTY, YELLOW],
[RED, EMPTY, EMPTY, EMPTY, BLUE]
```

Gambar 4. 26 Source Code 2

Baris 1 ([YELLOW, EMPTY, EMPTY, EMPTY, EMPTY]):

Baris pertama dari matriks.

- Elemen pertama (YELLOW) berada di kolom pertama.
- Kolom-kolom lainnya (2, 3, 4, 5) kosong.

Baris 2 ([EMPTY, EMPTY, EMPTY, EMPTY, EMPTY]):

Baris kedua dari matriks.

- Seluruh elemen pada baris ini adalah EMPTY, menunjukkan bahwa semua kotak pada baris ini kosong.

Baris 3 ([EMPTY, EMPTY, GREEN, EMPTY, EMPTY]): Baris ketiga dari matriks.

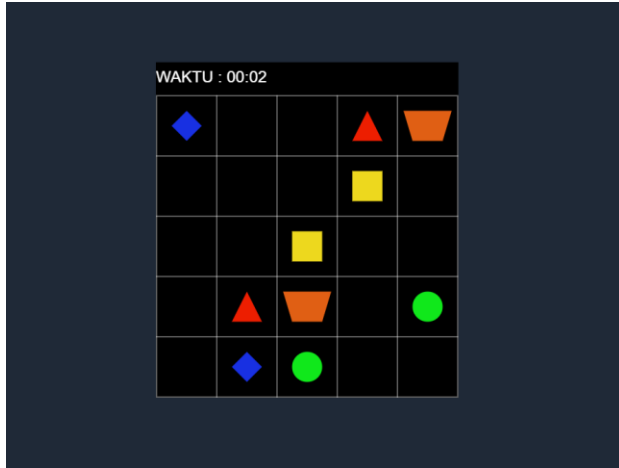
- Elemen ketiga (GREEN) berada di kolom ketiga.
- Kolom-kolom lainnya (1, 2, 4, 5) kosong.

Baris 4 ([BLUE, GREEN, RED, EMPTY, YELLOW]): Baris keempat dari matriks.

- Elemen pertama (BLUE) berada di kolom pertama.
- Elemen kedua (GREEN) berada di kolom kedua.
- Elemen ketiga (RED) berada di kolom ketiga.
- Kolom keempat (4) kosong.
- Elemen kelima (YELLOW) berada di kolom kelima.

Baris 5 ([RED, EMPTY, EMPTY, EMPTY, BLUE]): Baris kelima dari matriks.

- Elemen pertama (RED) berada di kolom pertama.
- Kolom-kolom lainnya (2, 3, 4) kosong.
- Elemen kelima (BLUE) berada di kolom kelima.



Gambar 4. 27 Permainan Flow 3

```
[
  [BLUE, EMPTY, EMPTY, RED, ORANGE],
  [EMPTY, EMPTY, EMPTY, YELLOW, EMPTY],
  [EMPTY, EMPTY, YELLOW, EMPTY, EMPTY],
  [EMPTY, RED, ORANGE, EMPTY, GREEN],
  [EMPTY, BLUE, GREEN, EMPTY, EMPTY]
],
```

Gambar 4. 28 Source Code 3

Baris 1 ([BLUE, EMPTY, EMPTY, RED, ORANGE]): Baris pertama dari matriks.

- Elemen pertama (BLUE) berada di kolom pertama.
- Kolom kedua (2) kosong.
- Kolom ketiga (3) kosong.
- Elemen keempat (RED) berada di kolom keempat.
- Elemen kelima (ORANGE) berada di kolom kelima.

Baris 2 ([EMPTY, EMPTY, EMPTY, YELLOW, EMPTY]):
Baris kedua dari matriks.

- Seluruh elemen pada baris ini adalah EMPTY, kecuali elemen ketiga (YELLOW) yang berada di kolom ketiga.

Baris 3 ([EMPTY, EMPTY, YELLOW, EMPTY, EMPTY]):
Baris ketiga dari matriks.

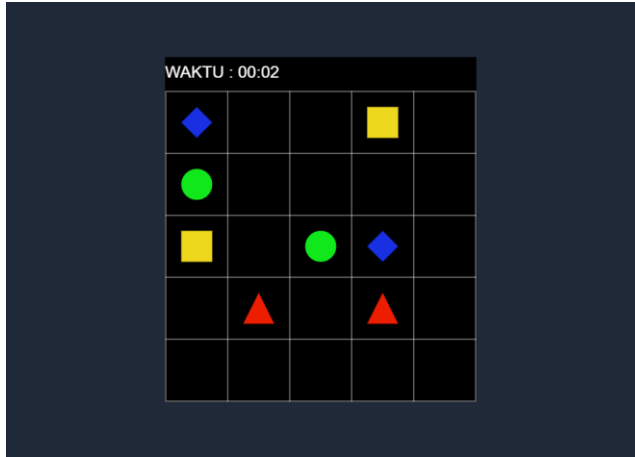
- Elemen ketiga (YELLOW) berada di kolom ketiga.
- Kolom-kolom lainnya (1, 2, 4, 5) kosong.

Baris 4 ([EMPTY, RED, ORANGE, EMPTY, GREEN]): Baris keempat dari matriks.

- Elemen kedua (RED) berada di kolom kedua.
- Elemen ketiga (ORANGE) berada di kolom ketiga.
- Kolom pertama (1) kosong.
- Elemen kelima (GREEN) berada di kolom kelima.

Baris 5 ([EMPTY, BLUE, GREEN, EMPTY, EMPTY]): Baris kelima dari matriks.

- Elemen kedua (BLUE) berada di kolom kedua.
- Elemen ketiga (GREEN) berada di kolom ketiga.
- Kolom keempat (4) kosong.
- Kolom kelima (5) kosong.



Gambar 4. 29 Permainan Flow 4

```
[BLUE, EMPTY, EMPTY, YELLOW, EMPTY],
[GREEN, EMPTY, EMPTY, EMPTY, EMPTY],
[YELLOW, EMPTY, GREEN, BLUE, EMPTY],
[EMPTY, RED, EMPTY, RED, EMPTY],
[EMPTY, EMPTY, EMPTY, EMPTY, EMPTY]
```

Gambar 4. 30 Source Code 4

Baris 1 ([BLUE, EMPTY, EMPTY, YELLOW, EMPTY]):

Baris pertama dari matriks.

- Elemen pertama (BLUE) berada di kolom pertama.
- Kolom kedua (2) kosong.
- Kolom ketiga (3) kosong.
- Elemen keempat (YELLOW) berada di kolom keempat.
- Kolom kelima (5) kosong.

Baris 2 ([GREEN, EMPTY, EMPTY, EMPTY, EMPTY]):
Baris kedua dari matriks.

- Elemen pertama (GREEN) berada di kolom pertama.
- Kolom-kolom lainnya (2, 3, 4, 5) kosong.

Baris 3 ([YELLOW, EMPTY, GREEN, BLUE, EMPTY]):
Baris ketiga dari matriks.

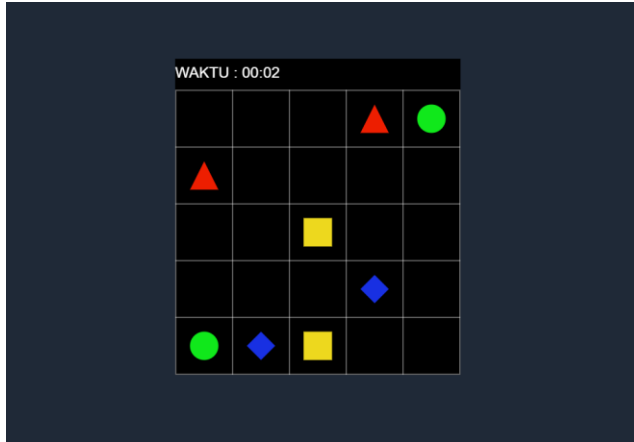
- Elemen pertama (YELLOW) berada di kolom pertama.
- Kolom kedua (2) kosong.
- Elemen ketiga (GREEN) berada di kolom ketiga.
- Elemen keempat (BLUE) berada di kolom keempat.
- Kolom kelima (5) kosong.

Baris 4 ([EMPTY, RED, EMPTY, RED, EMPTY]): Baris
keempat dari matriks.

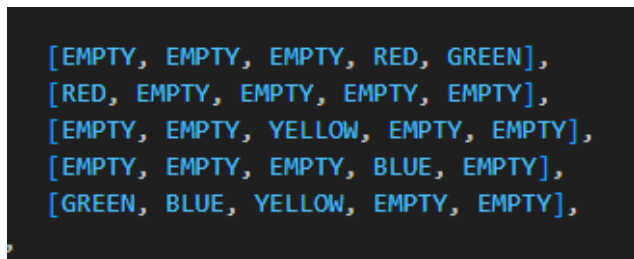
- Elemen kedua (RED) berada di kolom kedua.
- Kolom pertama (1) kosong.
- Elemen keempat (RED) berada di kolom keempat.
- Kolom ketiga (3) kosong.
- Kolom kelima (5) kosong.

Baris 5 ([EMPTY, EMPTY, EMPTY, EMPTY, EMPTY]):
Baris kelima dari matriks.

- Seluruh elemen pada baris ini adalah EMPTY, menunjukkan bahwa semua kotak pada baris ini kosong.



Gambar 4. 31 Permainan Flow 5



Gambar 4. 32 Source Code 5

Baris 1 ([EMPTY, EMPTY, EMPTY, RED, GREEN]): Baris pertama dari matriks.

- Kolom pertama (1) kosong.
- Kolom kedua (2) kosong.
- Kolom ketiga (3) kosong.
- Elemen keempat (RED) berada di kolom keempat.
- Elemen kelima (GREEN) berada di kolom kelima.

Baris 2 ([RED, EMPTY, EMPTY, EMPTY, EMPTY]): Baris kedua dari matriks.

- Elemen pertama (RED) berada di kolom pertama.
- Kolom-kolom lainnya (2, 3, 4, 5) kosong.

Baris 3 ([EMPTY, EMPTY, YELLOW, EMPTY, EMPTY]): Baris ketiga dari matriks.

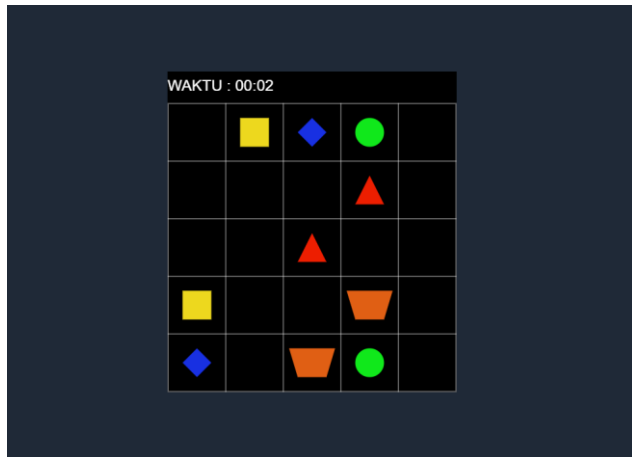
- Kolom pertama (1) kosong.
- Kolom kedua (2) kosong.
- Elemen ketiga (YELLOW) berada di kolom ketiga.
- Kolom keempat (4) kosong.
- Kolom kelima (5) kosong.

Baris 4 ([EMPTY, EMPTY, EMPTY, BLUE, EMPTY]): Baris keempat dari matriks.

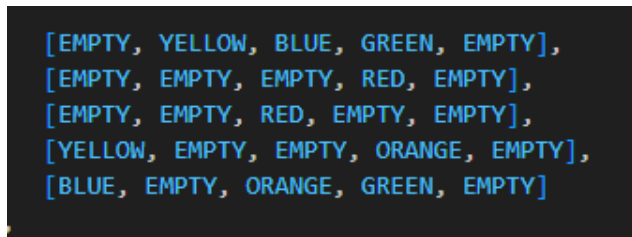
- Kolom pertama (1) kosong.
- Kolom kedua (2) kosong.
- Kolom ketiga (3) kosong.
- Elemen keempat (BLUE) berada di kolom keempat.
- Kolom kelima (5) kosong.

Baris 5 ([GREEN, BLUE, YELLOW, EMPTY, EMPTY]): Baris kelima dari matriks.

- Elemen pertama (GREEN) berada di kolom pertama.
- Elemen kedua (BLUE) berada di kolom kedua.
- Elemen ketiga (YELLOW) berada di kolom ketiga.
- Kolom keempat (4) kosong.
- Kolom kelima (5) kosong.



Gambar 4. 33 Permainan Flow 6



Gambar 4. 34 Source Code 6

Baris 1 ([EMPTY, YELLOW, BLUE, GREEN, EMPTY]):
Baris pertama dari matriks.

- Kolom pertama (1) kosong.
- Elemen kedua (YELLOW) berada di kolom kedua.
- Elemen ketiga (BLUE) berada di kolom ketiga.
- Elemen keempat (GREEN) berada di kolom keempat.
- Kolom kelima (5) kosong.

Baris 2 ([EMPTY, EMPTY, EMPTY, RED, EMPTY]): Baris kedua dari matriks.

- Seluruh elemen pada baris ini adalah EMPTY, kecuali elemen keempat (RED) yang berada di kolom keempat.

Baris 3 ([EMPTY, EMPTY, RED, EMPTY, EMPTY]): Baris ketiga dari matriks.

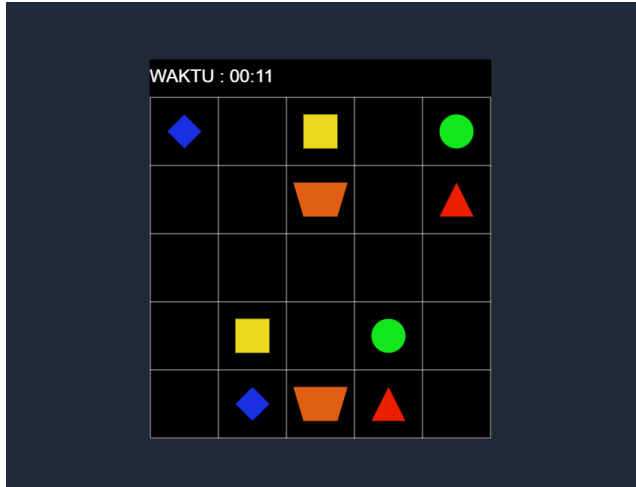
- Kolom pertama (1) kosong.
- Kolom kedua (2) kosong.
- Elemen ketiga (RED) berada di kolom ketiga.
- Kolom keempat (4) kosong.
- Kolom kelima (5) kosong.

Baris 4 ([YELLOW, EMPTY, EMPTY, ORANGE, EMPTY]): Baris keempat dari matriks.

- Elemen pertama (YELLOW) berada di kolom pertama.
- Kolom kedua (2) kosong.
- Kolom ketiga (3) kosong.
- Elemen keempat (ORANGE) berada di kolom keempat.
- Kolom kelima (5) kosong.

Baris 5 ([BLUE, EMPTY, ORANGE, GREEN, EMPTY]): Baris kelima dari matriks.

- Elemen pertama (BLUE) berada di kolom pertama.
- Kolom kedua (2) kosong.
- Elemen ketiga (ORANGE) berada di kolom ketiga.
- Elemen keempat (GREEN) berada di kolom keempat.
- Kolom kelima (5) kosong.



Gambar 4. 35 Permainan Flow 7

```
[BLUE, EMPTY, YELLOW, EMPTY, GREEN],
[EMPTY, EMPTY, ORANGE, EMPTY, RED],
[EMPTY, EMPTY, EMPTY, EMPTY, EMPTY],
[EMPTY, YELLOW, EMPTY, GREEN, EMPTY],
[EMPTY, BLUE, ORANGE, RED, EMPTY]
```

Gambar 4. 36 Source Code 7

Baris 1 ([BLUE, EMPTY, YELLOW, EMPTY, GREEN]):
Baris pertama dari matriks.

- Elemen pertama (BLUE) berada di kolom pertama.
- Kolom kedua (2) kosong.
- Elemen ketiga (YELLOW) berada di kolom ketiga.
- Kolom keempat (4) kosong.
- Elemen kelima (GREEN) berada di kolom kelima.

Baris 2 ([EMPTY, EMPTY, ORANGE, EMPTY, RED]): Baris kedua dari matriks.

- Kolom pertama (1) kosong.
- Kolom kedua (2) kosong.
- Elemen ketiga (ORANGE) berada di kolom ketiga.
- Kolom keempat (4) kosong.
- Elemen kelima (RED) berada di kolom kelima.

Baris 3 ([EMPTY, EMPTY, EMPTY, EMPTY, EMPTY]): Baris ketiga dari matriks.

- Seluruh elemen pada baris ini adalah EMPTY, menunjukkan bahwa semua kotak pada baris ini kosong.

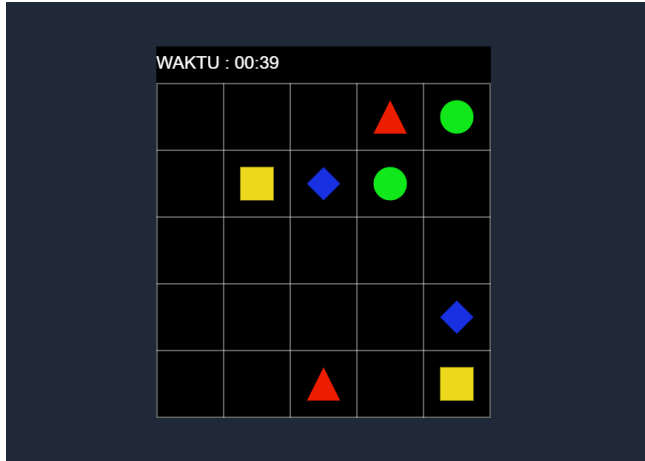
Baris 4 ([EMPTY, YELLOW, EMPTY, GREEN, EMPTY]): Baris keempat dari matriks.

- Kolom pertama (1) kosong.
- Elemen kedua (YELLOW) berada di kolom kedua.
- Kolom ketiga (3) kosong.
- Elemen keempat (GREEN) berada di kolom keempat.
- Kolom kelima (5) kosong.

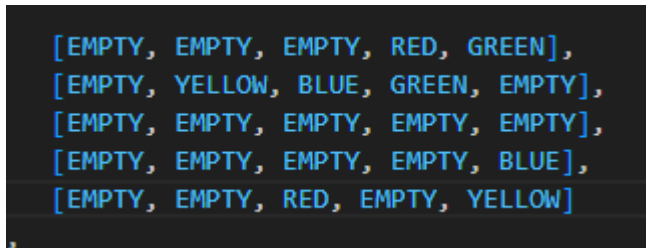
Baris 5 ([EMPTY, BLUE, ORANGE, RED, EMPTY]): Baris kelima dari matriks.

- Kolom pertama (1) kosong.
- Elemen kedua (BLUE) berada di kolom kedua.
- Elemen ketiga (ORANGE) berada di kolom ketiga.
- Elemen keempat (RED) berada di kolom keempat.

- Kolom kelima (5) kosong.



Gambar 4. 37 Permainan Flow 8



Gambar 4. 38 Source Code 8

Baris 1 ([EMPTY, EMPTY, EMPTY, RED, GREEN]): Baris pertama dari matriks.

- Seluruh kolom kecuali kolom keempat (RED) dan kolom kelima (GREEN) kosong.

Baris 2 ([EMPTY, YELLOW, BLUE, GREEN, EMPTY]): Baris kedua dari matriks.

- Elemen kedua (YELLOW) berada di kolom kedua.
- Elemen ketiga (BLUE) berada di kolom ketiga.
- Elemen keempat (GREEN) berada di kolom keempat.
- Seluruh kolom kecuali kolom kedua, ketiga, dan keempat kosong.

Baris 3 ([EMPTY, EMPTY, EMPTY, EMPTY, EMPTY]):

Baris ketiga dari matriks.

- Seluruh elemen pada baris ini adalah EMPTY, menunjukkan bahwa semua kotak pada baris ini kosong.

Baris 4 ([EMPTY, EMPTY, EMPTY, EMPTY, BLUE]):

Baris keempat dari matriks.

- Elemen kelima (BLUE) berada di kolom kelima.
- Seluruh kolom kecuali kolom kelima kosong.

Baris 5 ([EMPTY, EMPTY, RED, EMPTY, YELLOW]):

Baris kelima dari matriks.

- Elemen ketiga (RED) berada di kolom ketiga.
- Elemen kelima (YELLOW) berada di kolom kelima.
- Seluruh kolom kecuali kolom ketiga dan kelima kosong.

4.1.3 Penerapan Cara Kerja Permainan Flow

Berikut adalah langkah – langkah penerapan cara kerja permainan flow secara sistematis:

1. Langkah pertama adalah *login*, jika pengguna belum mempunyai akun bisa ke menu *Register*.

2. Bila pengguna telah menyelesaikan *Register* pengguna bisa langsung memasukkan *username* dan *password* lalu *login*.
3. Setelah masuk pengguna akan disuguhkan dengan salah satu level permainan flow.
4. Pengguna diharapkan menyelesaikan level permainan dengan benar.
5. Jika pengguna salah saat menyelesaikan akan mendapatkan level permainan terbaru, bila pengguna masih salah saat menyelesaikan level akan mendapatkan kesempatan 5 kali mengulang.
6. Jika masih salah sebanyak 5 kali pengguna akan mendapatkan notif *error* dan kembali ke halaman login.
7. Bila pengguna menyelesaikan dengan benar akan lanjut ke halaman *dashboard* yang didalamnya terdapat hasil tempuh waktu menyelesaikan.

Dengan mengikuti langkah – langkah ini, pengguna dapat dengan sistematis mengoperasikan permainan flow

4.1.4 Dashboard

Dashboard ini dirancang untuk memantau dan mencatat penyelesaian CAPTCHA menggunakan metode permainan flow. Berikut adalah penjelasan mengenai elemen-elemen yang ada pada dashboard:

1. Header:

Terdapat judul "Dashboard" di bagian atas halaman yang menandakan bahwa halaman ini adalah bagian dari sistem pengelolaan CAPTCHA.

2. User Information:

Bagian ini menampilkan informasi pengguna yang sedang login, yaitu nama lengkap pengguna dan durasi penyelesaian CAPTCHA. Full Name: Menampilkan nama pengguna, dalam contoh ini adalah "uwks". Durasi Penyelesaian CAPTCHA: Menampilkan durasi yang dibutuhkan oleh pengguna untuk menyelesaikan CAPTCHA, yang saat ini tercatat sebagai "00:00:00".

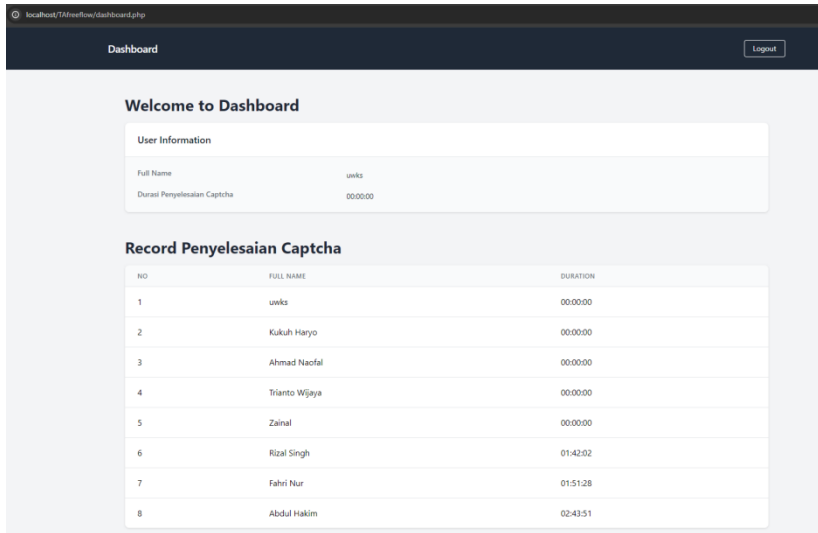
3. Record Penyelesaian Captcha:

Bagian ini mencatat dan menampilkan daftar penyelesaian CAPTCHA oleh berbagai pengguna. NO: Nomor urut penyelesaian CAPTCHA. Full Name: Nama lengkap pengguna yang menyelesaikan CAPTCHA. Duration: Durasi waktu yang dibutuhkan oleh masing-masing pengguna untuk menyelesaikan CAPTCHA. Dari data yang tercantum: Pengguna "uwks", "Kukuh Haryo", "Ahmad Naofal", "Trianto Wijaya", dan "Zainal" semuanya tercatat menyelesaikan CAPTCHA dengan durasi "00:00:00", yang mungkin menunjukkan percobaan atau data awal. Pengguna lain seperti "Rizal Singh", "Fahri Nur", dan "Abdul Hakim" memiliki durasi penyelesaian CAPTCHA masing-masing "01:42:02", "01:51:28", dan "02:43:51".

4. *Logout Button*:

Terdapat tombol "Logout" di pojok kanan atas halaman yang memungkinkan pengguna untuk keluar dari dashboard dan mengakhiri sesi mereka dengan aman. Dashboard ini memberikan gambaran yang jelas mengenai performa dan keterlibatan pengguna dalam menyelesaikan CAPTCHA

permainan flow, membantu administrator untuk memonitor dan menganalisis efektivitas sistem CAPTCHA yang diterapkan.



The screenshot shows a web dashboard with a dark header containing the text 'Dashboard' and a 'Logout' button. Below the header, there is a 'Welcome to Dashboard' section followed by a 'User Information' box. This box displays 'Full Name' as 'uwks' and 'Durasi Penyelesaian Captcha' as '00:00:00'. Below this is a section titled 'Record Penyelesaian Captcha' which contains a table with 8 rows of data. The table has three columns: 'NO', 'FULL NAME', and 'DURATION'.

NO	FULL NAME	DURATION
1	uwks	00:00:00
2	Kukuh Haryo	00:00:00
3	Ahmad Naofal	00:00:00
4	Trianto Wijaya	00:00:00
5	Zainal	00:00:00
6	Rizal Singh	01:42:02
7	Fahri Nur	01:51:28
8	Abdul Hakim	02:43:51

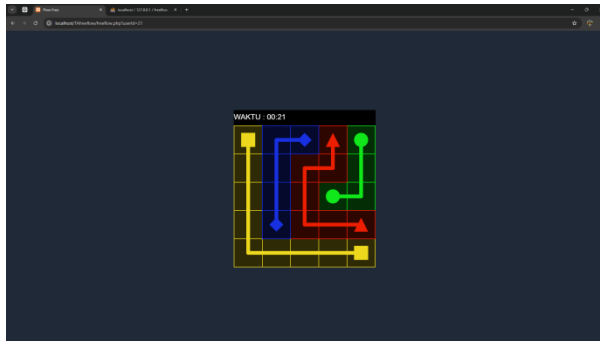
Gambar 4. 39 Dashboard

4.2 Hasil Penelitian

Penelitian ini bertujuan untuk merancang dan mengimplementasikan CAPTCHA permainan flow sebagai solusi inovatif dalam verifikasi manusia dan pengamanan sistem online. CAPTCHA permainan flow mengintegrasikan elemen permainan yang dirancang agar dapat diatasi oleh kemampuan kognitif manusia namun sulit diotomatisasi oleh bot. Berbagai tugas permainan, seperti puzzle, teka-teki, atau tugas interaktif lainnya, digunakan untuk menciptakan pengalaman pengguna yang positif sambil menyediakan lapisan keamanan tambahan.

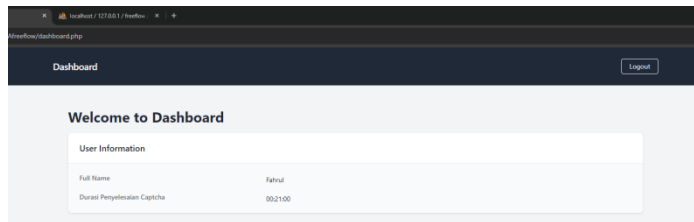
4.2.1 Testing Sistem pada *Audience*

1. User 1



Gambar 4. 40 Level User 1

Gambar diatas menunjukkan bahwa user 1 mendapatkan pola flow ke 4, user diharuskan untuk melakukan penyelesaian pola. Setelah itu user telah menyelesaikan dan menghubungkan semua titik dengan benar.



Gambar 4. 41 Hasil User 1

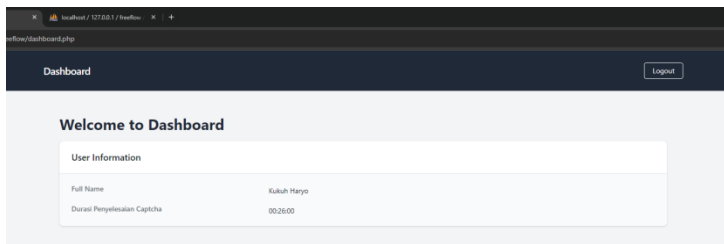
User menyelesaikan permainan dengan menempuh waktu 00:21 detik. Dengan waktu tersebut user telah menyelesaikan permainan dengan tuntas.

2. User 2



Gambar 4. 42 Level User 2

Gambar diatas menunjukkan bahwa user 2 mendapatkan pola flow ke 5, user diharuskan untuk melakukan penyelesaian pola. Setelah itu user telah menyelesaikan dan menghubungkan semua titik dengan benar.



Gambar 4. 43 Hasil User 2

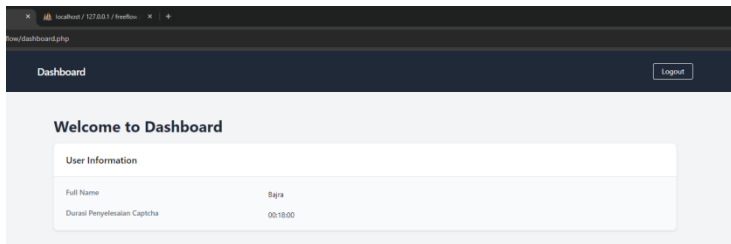
User menyelesaikan permainan dengan menempuh waktu 00:26 detik. Dengan waktu tersebut user telah menyelesaikan permainan dengan tuntas.

4. User 4



Gambar 4. 46 Level User 4

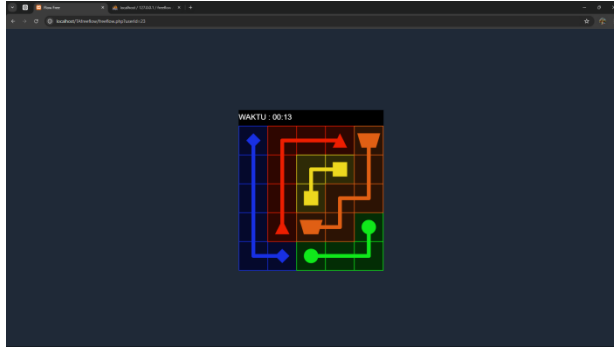
Gambar diatas menunjukkan bahwa user 4 mendapatkan pola flow ke 1, user diharuskan untuk melakukan penyelesaian pola. Setelah itu user telah menyelesaikan dan menghubungkan semua titik dengan benar.



Gambar 4. 47 Hasil User 5

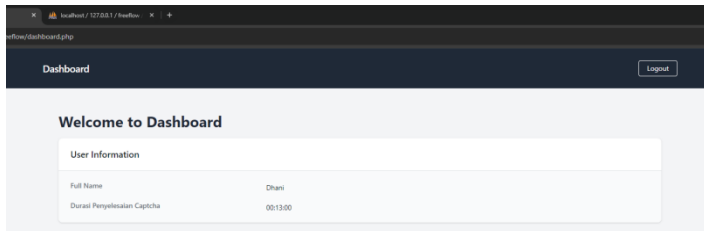
User menyelesaikan permainan dengan menempuh waktu 00:18 detik. Dengan waktu tersebut user telah menyelesaikan permainan dengan tuntas.

5. User 5



Gambar 4. 48 Level User 5

Gambar diatas menunjukkan bahwa user 5 mendapatkan pola flow ke 8, user diharuskan untuk melakukan penyelesaian pola. Setelah itu user telah menyelesaikan dan menghubungkan semua titik dengan benar.



Gambar 4. 49 Hasil User 5

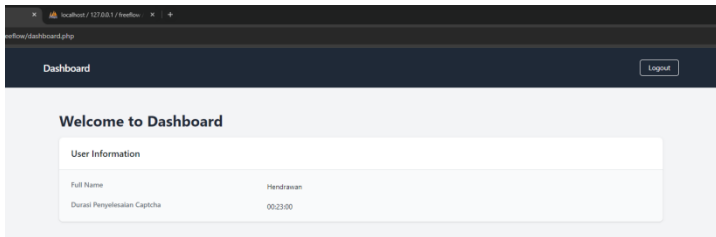
User menyelesaikan permainan dengan menempuh waktu 00:13 detik. Dengan waktu tersebut user telah menyelesaikan permainan dengan tuntas.

7. User 7



Gambar 4. 52 Level User 7

Gambar diatas menunjukkan bahwa user 7 mendapatkan pola flow ke 8, user diharuskan untuk melakukan penyelesaian pola. Setelah itu user telah menyelesaikan dan menghubungkan semua titik dengan benar.



Gambar 4. 53 Hasil User 7

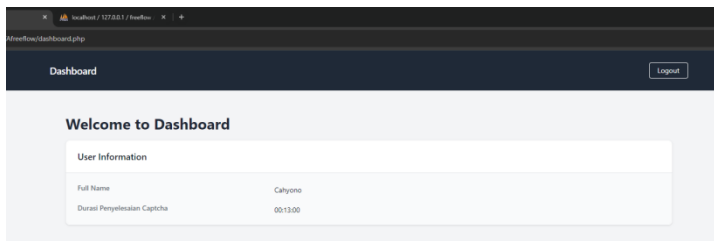
User menyelesaikan permainan dengan menempuh waktu 00:23 detik. Dengan waktu tersebut user telah menyelesaikan permainan dengan tuntas.

8. User 8



Gambar 4. 54 Level User 8

Gambar diatas menunjukkan bahwa user 8 mendapatkan pola flow ke 2, user diharuskan untuk melakukan penyelesaian pola. Setelah itu user telah menyelesaikan dan menghubungkan semua titik dengan benar.



Gambar 4. 55 Hasil User 8

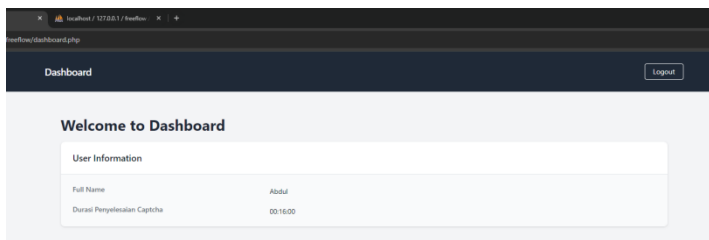
User menyelesaikan permainan dengan menempuh waktu 00:13 detik. Dengan waktu tersebut user telah menyelesaikan permainan dengan tuntas.

9. User 9



Gambar 4. 56 Level User 9

Gambar diatas menunjukkan bahwa user 9 mendapatkan pola flow ke 7, user diharuskan untuk melakukan penyelesaian pola. Setelah itu user telah menyelesaikan dan menghubungkan semua titik dengan benar.



Gambar 4. 57 Hasil User 9

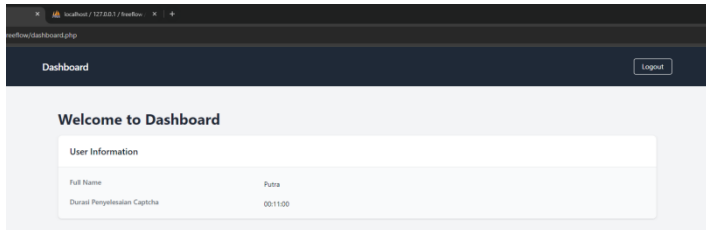
User menyelesaikan permainan dengan menempuh waktu 00:16 detik. Dengan waktu tersebut user telah menyelesaikan permainan dengan tuntas.

10. User 10



Gambar 4. 58 Level User 10

Gambar diatas menunjukkan bahwa user 10 mendapatkan pola flow ke 7, user diharuskan untuk melakukan penyelesaian pola. Setelah itu user telah menyelesaikan dan menghubungkan semua titik dengan benar.



Gambar 4. 59 Hasil User 10

User menyelesaikan permainan dengan menempuh waktu 00:11 detik. Dengan waktu tersebut user telah menyelesaikan permainan dengan tuntas.