

## **4. IMPLEMENTASI SISTEM**

### **4.1 Implementasi Perangkat Lunak yang Digunakan**

*Website* scan QR ini dibuat dengan menggunakan bahasa pemrograman PHP dengan menggunakan *framework* Laravel 8. Tampilan *front end* dibuat dengan menggunakan HTML, CSS, dan Bootstrap, sedangkan *back end* didukung sepenuhnya dengan Laravel 8. Database yang digunakan pada *website* ini adalah MySQL

### **4.2 Aplikasi Pemrograman**

Dalam Sub Bab ini akan dijelaskan mengenai aplikasi pemrograman yang telah digunakan untuk mengembangkan dan membuat aplikasi scan QR untuk pegawai pada CV. Indoteknik. Berikut ini adalah aplikasi yang digunakan :

- Visual Studio Code digunakan sebagai *code editor* dalam pengembangan *website*. Aplikasi ini digunakan pada semua tahap mulai dari awal pembuatan website hingga selesai.
- Composer digunakan saat melakukan instalasi laravel dan *package* yang diperlukan dalam pengembangan website. Composer ini digunakan dalam tahap awal pengembangan website
- Google Chrome merupakan perangkat lunak yang dapat berfungsi untuk mencari informasi yang ada di internet. Selain itu, google chrome juga dapat berfungsi untuk melakukan *debugging* dan menjalankan *code* dari website yang dibuat. Google Chrome juga dipakai untuk *view website* dan *debugging error* yang muncul.
- XAMPP merupakan *local server environment* yang digunakan untuk menjalankan *database management MySQL database*. XAMPP ini digunakan setiap kali untuk mengakses *database* yang menyimpan data yang dibutuhkan.
- Figma digunakan untuk merancang dari tampilan website yang akan dibuat. Hal ini dilakukan agar dapat melihat gambaran kasar dari user interface website. Figma digunakan dalam tahap awal dan menengah pengembangan website.

#### 4.3 Implementasi Sistem

Pada bab sebelumnya, telah dijelaskan mengenai desain sistem yang berisi fitur - fitur yang akan diimplementasikan pada website. Pembuatan program akan terbagi ke dalam bentuk segmen. Adapun pembagian segmen program akan dipetakan pada Tabel.

Tabel 4.1

Hubungan Desain dengan Segmen Program

Activity Diagram	Use Case	Segmen	Nama Segmen	Keterangan
-	-	4.1	Koneksi ke Database	Melakukan konfigurasi sistem untuk koneksi ke database
Gambar 3.2	Gambar 3.1	4.2	Login	Melakukan login pada user yang terdaftar
-	-	4.3	Menambah Pegawai	Menambah Pegawai dengan detail yang ada
-	-	4.4	Menambah Operator	Menambah Operator yang bekerja
-	-	4.5	Menambah Mesin	Menambah Mesin yang bisa digunakan
-	-	4.6	Menambah Purchase - Order	Menambah Pesanan yang bisa dilakukan oleh perusahaan
Gambar 3.3 Gambar 3.4 Gambar 3.5	-	4.7 4.8 4.9	Scan QR Barang Masuk	Fitur Scan QR untuk barang yang akan dikerjakan
-	-	4.10 4.11 4.12	Scan QR Barang Selesai	Fitur Scan QR untuk barang yang telah selesai diproses

#### 4.3.1 Koneksi ke Database

Untuk menghubungkan *framework* dari *directory* dengan *localhost*, serta menyambungkan sistem dengan *database*, diperlukan sebuah pengaturan koneksi. Hal yang dapat dilakukan adalah dengan mengatur file *.env* yang terdapat pada *file directory* tempat dimana laravel di *install*. *File* tersebut juga perlu dilakukan penyesuaian sehingga sesuai dengan pengaturan pada *database MySQL*

Segmen Program 4.1 Konfigurasi environment untuk koneksi ke database

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:CJuJYvYHTrHxiN7cR251d9oUkz/ZtUsgFKvl2TMSsHA=
APP_DEBUG=false
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_indotehnik1
DB_USERNAME=root
DB_PASSWORD=
```

#### 4.3.2 Login

Fitur login digunakan untuk melakukan pengecekan apakah user sudah terdaftar pada database yang ada. Melalui fitur ini, dilakukan pengecekan antara email dan password yang dimasukkan. Apabila cocok, maka user bisa masuk ke halaman dashboard sesuai hak akses yang mereka miliki. Source code dari login dapat dilihat pada segmen berikut ini.

Segmen Program 4.2 Source Code Login

```
public function login(Request $request)
{
    $this->validateLogin($request);

    if (method_exists($this, 'hasTooManyLoginAttempts') &&
        $this->hasTooManyLoginAttempts($request)) {
```

```

        $this->fireLockoutEvent($request);

        return $this->sendLockoutResponse($request);
    }

    if ($this->attemptLogin($request)) {
        if ($request->hasSession()) {
            $request->session()->put('auth.password_confirmed_at', time());
        }

        return $this->sendLoginResponse($request);
    }

    $this->incrementLoginAttempts($request);

    return $this->sendFailedLoginResponse($request);
}

```

#### 4.3.3 Menambah Pegawai

Pada segmen ini, user dengan role *super admin* dapat menambah pegawai melalui form yang telah disediakan dan setelah itu dapat disimpan di *database*.

##### Segmen Program 4.3 Source Code Tambah Pegawai

```

public function store(Request $request)
{
    $validator = Validator::make(
        $request->all(),
        [
            'kode' => 'required',
            'nama' => 'required',
            'NPWP' => 'required',
            'alamat' => 'required',
            'kota' => 'required',
            'kode_pos' => 'required',
            'no_telp' => 'required',
            'fax' => 'required',
            'email' => 'required',
            'kontak' => 'required',
            'password' => 'required',
        ],
    );
}

```

```

// dd("Proses Insert data", $request->all());
try {
    $masterPegawai = MasterPegawai::create([
        'kode' => $request->kode,
        'nama' => $request->nama,
        'NPWP' => $request->NPWP,
        'alamat' => $request->alamat,
        'kota' => $request->kota,
        'kode_pos' => $request->kode_pos,
        'no_telp' => $request->no_telp,
        'fax' => $request->fax,
        'email' => $request->email,
        'kontak' => $request->kontak,
        'password' => Hash::make($request->password)
    ]);
    Alert::success('Input Data', 'Input Data Berhasil');
    return redirect()->route('masterpegawai.index');
} catch (\Throwable $th) {
    Alert::error('Input Data', $th);
    return redirect()->back()->withInput($request->all())->withErrors($validator);
}
}

```

#### 4.3.4 Menambah Operator

Pada segmen ini, user dengan role *super admin* dapat menambah operator melalui form yang telah disediakan dan setelah itu dapat disimpan di *database*.

#### Segmen Program 4.4 Source Code Tambah Operator

```

public function store(Request $request)
{
    $dateValue = Carbon::now();
    $date = $dateValue->format('Y-m-d');
    $time = Carbon::now()->format('Y-m-d H:i:s');
    $validator = Validator::make(
        $request->all(),
        [
            'kode' => 'required',
            'nama' => 'required',
            'NPWP' => 'required',
            'alamat' => 'required',
        ]
    );
    if ($validator->fails()) {
        return redirect()->route('operator.create')
            ->withErrors($validator)
            ->withInput($request->all());
    }
    $operator = new Operator();
    $operator->fill($request->all());
    $operator->date = $date;
    $operator->time = $time;
    $operator->save();
    Alert::success('Operator', 'Operator Berhasil Ditambah');
    return redirect()->route('operator.index');
}

```

```

        'jenis_operator' => 'required',
    ],
);

try {
    $masterOperator = MasterOperator::create([
        'kode' => $request->kode,
        'nama' => $request->nama,
        'NPWP' => $request->NPWP,
        'alamat' => $request->alamat,
        'jenis_operator' => $request->jenis_operator
    ]);
    $historyoperator = [
        'tanggal' => $date,
        'nama_mesin' => $request->nama,
        'nama_barang' => $request->jenis_operator,
        'waktu' => $time,
        'keterangan' => 'Tambah',
    ];
    DB::table('history_operators')->insert($historyoperator);
    Alert::success('Input Data', 'Input Data Berhasil');
    return redirect()->route('masteroperator.index');
} catch (\Throwable $th) {
    Alert::error('Input Data', 'Ada Bagian yang Masih Kosong');
    return redirect()->back()->withInput($request->all())->withErrors($validator);
}
}
}

```

#### 4.3.5 Menambah Mesin

Pada segmen ini, user dengan role *super admin* dapat menambah mesin yang ada melalui form yang telah disediakan dan setelah itu dapat disimpan di *database*.

##### Segmen Program 4.5 Source Code Tambah Mesin

```

public function store(Request $request)
{
    $dt = Carbon::now();
    $todayDate = $dt->format('Y-m-d');
    $validator = Validator::make(
        $request->all(),
        [
            'kode' => 'required',

```

```

        'nama' => 'required',
        'keterangan' => 'required',
        'poin' => 'required',
    ],
);

try {
    MasterMesin::create([
        'kode' => $request->kode,
        'nama' => $request->nama,
        'keterangan' => $request->keterangan,
        'poin' => $request->poin,
    ]);
    $historymesin = [
        'tanggal' => $todayDate,
        'nama_mesin' => $request->nama,
        'nama_operator' => $request->jenis_operator,
        'poin' => $request->poin,
        'keterangan' => 'Tambah',
    ];
    DB::table('history_mesins')->insert($historymesin);
    Alert::success('Input Data', 'Input Data Berhasil');
    return redirect()->route('mastermesin.index');
} catch (\Throwable $th) {
    Alert::error('Input Data', 'Ada Bagian yang Masih Kosong' . $th);
    return redirect()->back()->withInput($request->all())->withErrors($validator);
}
}
}

```

#### 4.3.6 Menambah Purchase - Order

Pada segmen ini, user dengan role *super admin* dapat menambah surat *purchase - order* melalui form yang telah disediakan dan setelah itu dapat disimpan di *database*.

##### Segmen Program 4.6 Source Code Tambah Surat Purchase - Order

```

public function store(Request $request)
{
    $validator = Validator::make(
        $request->all(),
        [
            'nomor' => 'required',
            'tanggal' => 'required',

```

```

'customer' => 'required',
'quantity' => 'required',
'satuan' => 'required',
'explanation' => 'required',
'keterangan' => 'required',
],
);

try {
$masterPreorder = MasterPreorder::create([
'nomor' => $request->nomor,
'tanggal' => $request->tanggal,
'customer' => $request->customer,
'quantity' => $request->quantity,
'satuan' => $request->satuan,
'explanation' => $request->explanation,
'keterangan' => $request->keterangan
]);
Alert::success('Input Data', 'Input Data Berhasil');
return redirect()->route('masterpreorder.index');
} catch (\Throwable $th) {
Alert::error('Input Data', 'Ada Bagian yang Masih Kosong');
return redirect()->back()->withInput($request->all())->withErrors($validator);
}
}
}

```

#### 4.3.7 Scan QR Barang Masuk

Pada segmen ini, user dapat melakukan Scan QR atau memilih data yang ada pada dropdown untuk memasukkan data. Data tersebut terdiri dari nama operator dan mesin yang akan digunakan untuk menyelesaikan surat order.

##### Segmen Program 4.7 Source Code Submit Operator dan Mesin

```

$('#SubmitQR').click(function(e) {
var namaOperator = $('#nama_operator').val();
var namaMesin = $('#nama_mesin').val();
if (namaOperator != '' && namaMesin != '') {

    var history = "";
    for (var i = 0; i < operator.length; i++) {
        if (operator[i].nama == namaOperator) {
            history = operator[i].surat;
        }
    }
    if (history != "") {
        alert("Surat berhasil dikirim");
        window.location.href = "http://127.0.0.1:8000/masterpreorder/index";
    } else {
        alert("Surat gagal dikirim");
    }
}
});
```

```

        }
    }
    var pesan = "Nama: " + namaOperator + " Mesin: " + namaMesin;
    if (history != "") {
        pesan = "Nama: " + namaOperator + " (" + history + ") Mesin: " + namaMesin;
    }
    Swal.fire({
        title: 'Input Sudah Benar?',
        text: pesan,
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',
        cancelButtonColor: '#d33',
        confirmButtonText: 'Benar'
    }).then((result) => {
        if (result.isConfirmed) {
            $('#surat_order').prop('disabled', false)
            $('#inputGroupFile04').prop('disabled', false)
            $('#exampleFormControlTextarea1').prop('disabled', false)
            $('#nama_operator').prop('disabled', true)
            $('#nama_mesin').prop('disabled', true)
        } else {
            $('#surat_order').prop('disabled', true)
            $('#inputGroupFile04').prop('disabled', true)
            $('#exampleFormControlTextarea1').prop('disabled', true)
            $('#nama_operator').prop('disabled', false)
            $('#nama_mesin').prop('disabled', false)
        }
    })
} else {
    Swal.fire(
        'Lupa Input?',
        'Nama Operator atau Mesin masih Kosong',
        'question'
    )
}
});

```

Setelah melakukan Submit dengan data yang benar, maka user akan melakukan input lagi untuk surat, keterangan pengkerjaan, dan foto barang.

#### Segmen Program 4.8 Source Code Tambah Data

```

$('#tambah').click(function(e) {

```

```

if ($('#surat_order').val() != '-' && $('#exampleFormControlTextarea1').val() != '') {
    var suratCheck = null;
    for (var i = 0; i < surat.length; i++) {
        if (surat[i].nomor == $('#surat_order').val()) {
            suratCheck = surat[i];
        }
    }
    $('#BodyQR').append('<tr id=' + "IDX" + Index + '> <td>' + $('#nama_operator')
        .val() +
        '</td> <td>' + $('#nama_mesin').val() + '</td> <td>' + $('#surat_order')
        .val() +
        '</td> <td>' + today + '</td> <td>' + $(
            '#exampleFormControlTextarea1').val() +
        '</td> <td> <button type="button" class="btn btn-danger" onclick="DeleteRow('
+
        "IDX" + Index++ + ')"><i class="fas fa-trash"></i></button> </td><td>' + $(
            '#picture').val() + '</td></tr>');
    }

} else {
    Swal.fire(
        'Lupa Input?',
        'Ada Salah Satu Bagian Masih Kosong',
        'question'
    )
}
});

```

Setelah melakukan penambahan data, maka data yang ada akan dimasukan kedalam tabel sementara yang dimana fungsinya untuk menampung semua surat yang akan dikerjakan di mesin tersebut. Setelah melakukan pengecekan, maka data bisa dimasukkan ke dalam database dan pegawai bisa mengerjakan tugasnya.

#### Segmen Program 4.9 Source Code Submit Database QR

```

$('#SubmitDB').click(function(e) {
    var _token = $("input[name='_token']").val();
    var N_OP = []
    var N_MSN = []
    var N_ORD = []
    var Tanggal = []
    var Tanggal_K = []
    var Ket_P = []
    var picture = [];

```

```

$( "table > tbody > tr" ).each(function() {
    N_OP.push($(this).find('td').eq(0).text())
    N_MSN.push($(this).find('td').eq(1).text())
    N_ORD.push($(this).find('td').eq(2).text())
    Tanggal.push($(this).find('td').eq(3).text())
    Ket_P.push($(this).find('td').eq(4).text())
    picture.push($(this).find('td').eq(6).text())
});

console.log("picture", picture);
$.ajax({
    type: 'GET',
    url: '{{ route('SubmitQR') }}',
    data: {
        _token: _token,
        N_OP: N_OP,
        N_MSN: N_MSN,
        N_ORD: N_ORD,
        Tanggal: Tanggal,
        Tanggal_K: Tanggal_K,
        Ket_P: Ket_P,
        operator: $("#"nama_operator").val(),
        picture
    },
}).done(function(msg) {
    setTimeout(() => {
        window.location.href = "{{ route('QRMaster.barangMasuk.index') }}";
    }, 0);
})
});
});

```

#### 4.3.8 Scan QR Barang Selesai

Pada Segmen ini, barang yang telah selesai dikerjakan dapat dilakukan scan di halaman Scan QR Barang Selesai. Dengan begitu, data yang ada pada database akan ditandai selesai sehingga bisa menuju tabel selesai. Untuk menyelesaikan barang, perlu memasukkan penanggung jawab.

#### Segmen Program 4.10 Source Code Submit Penanggung Jawab

```

$('#SubmitQR').click(function(e) {
    var nama_pj = $('#nama_pj').val();
    // var namaMesin = $('#nama_mesin').val();

```

```

if (nama_pj != '-') {

    var history = "";
    for (var i = 0; i < operator.length; i++) {
        if (operator[i].nama == nama_pj) {
            history = operator[i].surat;
        }
    }
    var pesan = "Nama: " + nama_pj;
    if (history != "") {
        pesan = "Nama: " + nama_pj;
    }
    Swal.fire({
        title: 'Input Sudah Benar?',
        text: pesan,
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',
        cancelButtonColor: '#d33',
        confirmButtonText: 'Benar'
    }).then((result) => {
        if (result.isConfirmed) {
            $('#surat_order').prop('disabled', false)
            $('#inputGroupFile04').prop('disabled', false)
            $('#exampleFormControlTextarea1').prop('disabled', false)
            $('#nama_pj').prop('disabled', true)
        } else {
            $('#surat_order').prop('disabled', true)
            $('#inputGroupFile04').prop('disabled', true)
            $('#exampleFormControlTextarea1').prop('disabled', true)
            $('#nama_pj').prop('disabled', false)
        }
    })
} else {
    Swal.fire(
        'Lupa Input?',
        'Nama Operator atau Mesin masih Kosong',
        'question'
    )
}
});

```

Setelah melakukan Submit dengan data yang benar, maka user akan melakukan input lagi untuk surat, keterangan penggerjaan, dan foto barang.

#### Segmen Program 4.11 Source Code Tambah Data

```
$('#tambah').click(function(e) {
    if ($('#surat_order').val() != '-' && $('#exampleFormControlTextarea1').val() != '') {
        var suratCheck = null;
        for (var i = 0; i < surat.length; i++) {
            if (surat[i].nomor == $('#surat_order').val()) {
                suratCheck = surat[i];
            }
        }
        $('#BodyQR').append('<tr id=' + "IDX" + Index + '><td>' + $('#nama_pj')
            .val() +
            '</td><td>' + $('#surat_order')
            .val() +
            '</td><td>' + today + '</td><td>' + $('#exampleFormControlTextarea1').val() +
            '</td><td> <button type="button" class="btn btn-danger" onclick="DeleteRow('
+
            "IDX" + Index++ + ')"><i class="fas fa-trash"></i></button> </td><td>' + $('#picture').val() + '</td> </tr>');
    } else {
        Swal.fire(
            'Lupa Input?',
            'Ada Salah Satu Bagian Masih Kosong',
            'question'
        )
    }
});
```

Setelah melakukan penambahan data, maka data yang ada akan dimasukan kedalam tabel sementara yang dimana fungsinya untuk menampung semua surat yang telah selesai dikerjakan. Setelah melakukan pengecekan, maka data bisa submit dan penggerjaan barang telah selesai.

#### Segmen Program 4.12 Source Code Submit Database Selesai Scan QR

```
$('#SubmitDB').click(function(e) {
    var _token = $("input[name='_token']").val();
    var N_PJ = []
    var N_ORD = []
    var tgl_selesai = []
    var ket = []
    var picture = [];
    $("table > tbody > tr").each(function() {
```

```

N_PJ.push($(this).find('td').eq(0).text())
N_ORD.push($(this).find('td').eq(1).text())
tgl_selesai.push($(this).find('td').eq(2).text())
ket.push($(this).find('td').eq(3).text())
picture.push($(this).find('td').eq(5).text())
});

console.log("picture", picture);
$.ajax({
    type: 'GET',
    url: '{{ route('QRMaster.poSelesai.store') }}',
    data: {
        _token: _token,
        N_PJ: N_PJ,
        N_ORD: N_ORD,
        tgl_selesai: tgl_selesai,
        ket: ket,
        picture: picture,
    },
}).done(function(msg) {
    setTimeout(() => {
        window.location.href = "{{ route('QRMaster.poSelesai.index') }}";
    }, 0);
})
});
});

function DeleteRow(Row) {
    Row.innerHTML =
}

```