



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Praktik kerja magang dilaksanakan di PT Polaritas Multitrans Technology pada divisi IT di bawah bimbingan Bapak Ali Faturachman selaku Kepala Divisi IT. Pekerjaan atau kegiatan yang diberikan oleh perusahaan pada umumnya serupa dengan staff perusahaan. Mahasiswa diwajibkan mampu mengikuti prosedur yang ada di dalam perusahaan tersebut.

Karena ada projek baru yang ingin dibuat, pekerjaan yang diberikan pembimbing magang sesuai dengan kemampuan dan bidang yang dikuasai mahasiswa magang, yaitu membuat *user interface* pada sebuah aplikasi web untuk bagian *Inventory* dan membantu *back-end development* untuk mengkoneksikan database dengan aplikasi web.

Membuat *user interface* disini adalah membuat tampilan untuk sebuah aplikasi web agar mudah digunakan dan indah untuk dilihat oleh *user*. *User interface*

User interface yang akan dibuat selama program magang berlangsung adalah membuat tampilan untuk aplikasi web bagian *Inventory*. Menurut Bapak Ali Faturachman, aplikasi web pada bagian *Inventory* hanya akan dipakai oleh divisi *Warehouse*, sehingga tidak perlu membuat tampilan yang indah untuk dilihat tetapi mementingkan fungsionalitas dan kemudahan dalam penggunaan aplikasi web.

Aktivitas lain yang dilakukan selama melaksanakan praktik kerja magang yaitu membantu atau ikut serta dalam menyelesaikan project yang tidak berhubungan dengan divisi IT yaitu menginput data sertifikasi alat kesehatan serta belajar mengkalibrasi alat kesehatan yang ada di RS Eka Hospital, Tangerang Selatan.

Saat belajar mengkalibrasi alat kesehatan, bukan dibimbing oleh Bapak Ali Faturochman melainkan dibimbing oleh Bapak Agung selaku teknisi senior di PT Polaritas Multitrans Technology.

3.2 Tugas yang Dilakukan

Pada program kerja magang di PT Polaritas Multitrans Technology, diberikan beberapa tugas dan tanggung jawab yang harus dilakukan selama proses magang berlangsung. Hal yang pertama dilakukan adalah proses pengenalan dan pembekalan pengetahuan tentang jasa yang ditawarkan PT Polaritas Multitrans Technology sebagai perusahaan penyedia jasa kalibrasi alat kesehatan rumah sakit dan puskesmas.

Setelah memahami nilai perusahaan dan jasa yang ditawarkan, diberikan tugas pertama untuk membuat *user interface* untuk sistem aplikasi pada bagian *Inventory* dikarenakan kurangnya personil yang tersedia untuk membuat *user interface* pada bagian *Inventory*. Pembuatan *user interface* menggunakan sebuah *software* sehingga dapat menambah elemen atau fitur yang akan ada dalam *user interface* tersebut.

Tugas selanjutnya yang diberikan adalah membantu tim *back-end development* dalam mengkoneksikan *database* ke sistem aplikasi yang sedang dibuat. Tugas ini diberikan atas permintaan dari tim *back-end development* karena *deadline* yang diberikan untuk menyelesaikan *back-end development* mendekati jatuh tempo yang sudah ditentukan.

Tugas terakhir yang diberikan dan dapat dilakukan selama proses kerja magang berlangsung adalah ikut dalam kegiatan operasional perusahaan. Kegiatan operasional yang dimaksud adalah ikut dalam kegiatan kalibrasi alat – alat kesehatan rumah sakit dan melakukan *input data* seputar alat yang sudah dikalibrasi dan sertifiikat untuk alat yang sudah dikalibrasi tersebut.

3.3 Uraian Pekerjaan Magang

3.3.1 Pengenalan Lingkungan Kerja (Minggu 1 – 2)

Pada minggu pertama masa magang, tidak banyak melakukan kegiatan, hanya diperkenalkan ke setiap divisi yang ada di PT Polaritas Multitrans Technology terutama divisi IT agar dapat beradaptasi dengan cara kerja divisi tersebut. Divisi IT terdiri dari 1 kepala divisi dan 3 staf IT yang bertugas untuk mengurus untuk melakukan input data sertifikasi data alat kesehatan, melakukan maintenance terhadap sistem aplikasi yang sedang dijalankan oleh perusahaan, dan melakukan maintenance terhadap komputer dan jaringan yang dipakai oleh perusahaan.

Pada minggu kedua bapak Ali Faturochman selaku ketua divisi IT memberitahukan apa saja yang harus ada di dalam *User Interface* untuk *me-manage* Inventory serta membantu dalam *back-end development* untuk mengkoneksikan database dengan sistem aplikasi yang sedang dibuat oleh tim *Development* serta pengenalan tentang *software* yang akan digunakan untuk merancang *user interface* dan membantu *back-end development*.

Dalam membuat *user interface*, Bapak Ali Faturochman merekomendasikan untuk membuat tampilan yang mementingkan kemudahan dalam menggunakan dibandingkan keindahan *user interface*, dikarenakan aplikasi web yang sedang dibuat oleh perusahaan hanya akan dipakai oleh pihak internal PT Polaritas Multitrans Technology dimana para karyawannya lebih mementingkan kemudahan dalam menggunakan aplikasi web. Aplikasi yang digunakan dalam pembuatan *user interface* selama masa magang adalah Axure RP 9. Program ini merupakan salah satu *software* untuk merancang dan membuat *user interface* baik untuk *desktop* maupun *mobile*. Program ini dipilih dikarenakan kemudahan yang diberikan untuk membuat *user interface* yang interaktif namun mudah untuk digunakan.

Untuk membantu tim *back-end development*, Bapak Ali Faturochman merekomendasikan untuk membantu menghubungkan database dengan sistem aplikasi yang sedang dibuat. Database yang

akan dihubungkan berisikan data – data *inventory* dari PT Polaritas Multitrans Technology. Program yang digunakan dalam membantu *back-end development* adalah XAMPP dan Notepad++.

Notepad++ merupakan program pengolah kata yang dapat di *download* secara gratis untuk membuat file yang berisikan *coding* yang akan dibaca oleh server sehingga dapat melakukan apa yang file *coding* tersebut minta. Notepad++ dapat mengolah file dalam format *php*, *html*, *css*, dan masih banyak lagi format yang didukung oleh aplikasi ini.

XAMPP merupakan salah satu perangkat lunak yang dapat digunakan untuk memulai *server Apache* untuk melakukan *debug* pada *file* berbasis *php*, memulai *phpMyAdmin*, sebuah aplikasi berbasis web yang berfungsi untuk melihat database, melihat dan mengubah isi tabel dalam database, hingga menjadi *server* untuk melakukan pengetesan terhadap *coding* menghubungkan database dengan sistem aplikasi. Menggunakan perangkat lunak seperti XAMPP lebih mudah dan aman dalam melakukan *coding* yang membutuhkan database dikarenakan database yang akan digunakan tersimpan ke dalam penyimpanan komputer / notebook dari tim *back-end development*, sehingga jika server perusahaan sedang dalam masa *maintenance* atau ingin memeriksa apakah *coding* yang telah dibuat mampu menghubungkan database ke sistem aplikasi, tim *back-end development* dengan mudah menjalankan perangkat

lunak seperti XAMPP dan menyalakan server pribadi untuk langsung melakukan *testing* terhadap *coding* yang telah dibuat sebelumnya.

3.3.1 Membantu Back-end Development (Minggu ke 3 – 6)

Minggu ke 3 sampai minggu ke 6, diberikan tugas oleh Bapak Ali Faturachman untuk membantu tim *back-end development* dalam menghubungkan database dengan *user interface* yang telah ada. Tugas ini diberikan atas permintaan tim *back-end development* untuk tambahan personil dikarenakan *timeline* yang sedang berjalan sudah mendekati tanggal jatuh tempo. Kemudian berbincang – bincang dengan anggota dari tim Software yang digunakan dalam *back-end development* ini adalah XAMPP untuk melihat isi database yang digunakan, dan untuk melakukan *coding* menggunakan *software* Notepad++.

```

require '../koneksi.php';
$pdo = koneksiDb();

if($_GET['action'] == "add"){
    $sql = "INSERT INTO barang(idbarang, nama, foto)
        VALUES(?, ?, ?)";
    $stmt = $pdo->prepare($sql);

    $ext_boleh = ["jpg", "png", "jpeg"];
    if(checkFile($_FILES['foto'], $ext_boleh)){

        $ext = getFileExt($_FILES['foto']);
        $temp = $_FILES['foto']['tmp_name'];
        $foto_path = "../assets/images/user/" . $_POST['nama'] . "." . $ext;
        $stampil_foto_path = "assets/images/user/" . $_POST['nama'] . "." . $ext;
        if (!file_exists("../assets/images/user")) {
            mkdir("../assets/images/user");
        }
        move_uploaded_file($temp, $foto_path);

        $data = [
            $_POST['idbarang'],
            $_POST['nama'],
            $stampil_foto_path
        ];
    }else{
        $data = [
            $_POST['idbarang'],
            $_POST['nama'],
            null
        ];
    }
    $stmt->execute($data);

    header('Location: ../index.php?page=user');
}elseif ($_GET['action'] == "edit") {
    $sql = "UPDATE barang
        SET idbarang = ?,
            nama = ?,
            foto = ?
        WHERE id = ?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([
        $_POST['idbarang'],
        $_POST['nama'],
        $stampil_foto_path,
        $_POST['id']
    ]);
}

```

Gambar 3.1 Coding untuk back-end


Pada gambar 3.1 merupakan *coding* yang dibuat untuk mengkoneksikan database dengan sistem aplikasi yang dibuat. Coding ini dibuat berdasarkan ilmu yang didapat pada matakuliah Pemrograman Web di Universitas Multimedia Nusantara.


```
function koneksiDb(){
    try{
        $dsn = "mysql:host=localhost;dbname=inventory";
        $pdo = new PDO($dsn, 'root', '');
        return $pdo;
    }catch(PDOException $e){
        return $e;
    }
}
```

Gambar 3.2 Koneksi Database

Mengkoneksikan database dimulai dengan membuat *coding* untuk memulai koneksi ke server dan database yang akan digunakan. Pada gambar 3.2 koneksi database dilakukan menggunakan *class* PDO. *Class* ini dimulai dari memanggil fungsi *try* untuk mencoba membuat koneksi ke database pada *variable* \$dsn. Kemudian membuat *variable* \$pdo untuk menerapkan *class* PDO tersebut dengan parameter alamat server yang dituju, database yang dipilih, *username*, dan *password*. Fungsi *try* ini diakhiri dengan mengembalikan nilai dari \$pdo. Selama fungsi *try* berlangsung, ada fungsi *catch* yang berguna untuk mendapatkan *error* yang terjadi. Ketika *class* PDO dijalankan.

Dalam melakukan *coding* untuk menghubungkan database ke sistem aplikasi yang dibuat, harus melihat terlebih dahulu *table* yang ada di dalam database yang dipilih.

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id 	int(11)			No	None		AUTO_INCREMENT
id_alat	int(50)			No	None		
nama_alat	varchar(100)	utf8mb4_general_ci		No	None		
foto_alat	varchar(200)	utf8mb4_general_ci		No	None		

Gambar 3.3 Tabel Alat Kalibrasi

Pada gambar 3.3 adalah salah satu table dari *database* yang akan digunakan oleh sistem aplikasi. Tabel dinamakan *alat_kalibrasi* dan memiliki 4 kolom, kolom *id* berisikan nomor yang dimulai dari angka 1 hingga seterusnya dengan *value* berupa *integer*. *Value* pada kolom ini dibuat secara otomatis menggunakan fitur *Auto Increment* sehingga jika ada alat baru yang ditambahkan maka kolom *id* ini akan menentukan secara otomatis nomor untuk alat kalibrasi tersebut.

Kolom kedua berisikan nomor id dari alat kalibrasi yang didapatkan dari sistem penomoran alat yang dimiliki oleh divisi *warehouse* dengan *value* berupa *integer* maksimal 50 karakter. Perbedaan kolom ini dengan kolom *id* adalah pada kolom *id* digunakan dalam mempermudah *coding* agar ketika *user* memilih alat kalibrasi, maka dapat ditampilkan informasi tentang alat kalibrasi tersebut tanpa menambah *coding* yang akan menghambat *back-end development* lebih jauh.

Kolom ketiga yaitu nama_alat berisikan nama dari alat kalibrasi yang didaftarkan dengan *value* berupa *varchar* dengan 100 karakter maksimal.

Dan kolom terakhir dalam tabel ini adalah foto_alat yang berisikan lokasi foto dari alat kalibrasi tersebut dengan *value* berupa *varchar* dengan maksimal karakter 200. kolom foto_alat hanya berisikan lokasi dari foto alat kalibrasi karena ada *coding* yang akan membaca lokasi dari foto alat kalibrasi dan menampilkan nya ke *user*.

Setelah melihat isi dari database yang akan digunakan, selanjutnya membuat beberapa baris *coding* untuk melakukan fungsi CRUD (*Create, Read, Update, Delete*).

```

$sql = "INSERT INTO alat_kalibrasi(id, id_alat, nama_alat, foto_alat)
VALUES(?, ?, ?, ?)";
$stmt = $pdo->prepare($sql);

$ext_boleh = ["jpg", "png", "jpeg"];
if(checkFile($_FILES['foto_alat'], $ext_boleh)){

    $ext = getFileExt($_FILES['foto_alat']);
    function checkFile($file, Array $ext_boleh, $max_size = 0){
        $ext = getFileExt($file);
        if (in_array($ext, $ext_boleh) && $file['size'] <= $max_size || $max_size == 0) {
            return true;
        }else{
            return false;
        }
    }

    $foto_path = "../assets/images/alat/" . $_POST['nama_alat'] . "." . $ext;
    $stampil_foto_path = "assets/images/alat/" . $_POST['nama_alat'] . "." . $ext;
    if (!file_exists("../assets/images/alat")) {
        mkdir("../assets/images/alat");
    }
    move_uploaded_file($temp, $foto_path);

    $data = [
        $_POST['id'],
        $_POST['id_alat'],
        $_POST['nama_alat'],
        $stampil_foto_path
    ];
}else{
    $data = [
        $_POST['id'],
        $_POST['id_alat'],
        $_POST['nama_alat'],
        null
    ];
}
$stmt->execute($data);

```

Gambar 3.4 Coding untuk fungsi Create

Fungsi *Create* disini adalah fungsi untuk menambahkan data tentang alat kalibrasi baru ke dalam database. Pada gambar 3.4 fungsi *Create* dimulai dengan mendefinisikan variabel *\$sql* yang berisikan tugas berupa memasukkan *entry* ke dalam tabel dengan *values* yang berasal dari input yang *user* berikan, kemudian menyimpan variabel tersebut ke dalam *prepared statement* dikarenakan ada proses untuk menyimpan foto ke dalam database. Menyimpan foto ke dalam database bukan berarti menyimpan file foto dalam format *jpg* atau *png*, melainkan menyimpan lokasi foto

tersebut dalam format *String*. Proses ini dimulai dari membuat *variabel* `$ext` untuk mengambil format ekstensi foto seperti *jpg* atau *png*. Ini diperlukan karena kita akan mengubah *filename* foto menjadi nama alat kalibrasi berdasarkan *input* yang diberikan *user*.

```
function checkFile($file, Array $ext_boleh, $max_size = 0){
    $ext = getFileExt($file);

    if (in_array($ext, $ext_boleh) && $file['size'] <= $max_size || $max_size == 0) {
        return true;
    }else{
        return false;
    }
}
```

Gambar 3.5 Coding untuk filter foto

Selanjutnya membuat fungsi *checkFile* pada gambar 3.5 untuk memeriksa apakah foto yang dikirim sesuai dengan kriteria ukuran *file* yang ditentukan oleh tim *back-end development* dan database yang digunakan. Dikarenakan foto yang memiliki ukuran besar dapat memakan *bandwith* dan *storage* pada database yang lebih besar. Jika database sudah penuh maka divisi IT harus mengeluarkan uang untuk menambah kapasitas penyimpanan dimana akan berdampak pada peningkatan pengeluaran perusahaan.

```

$foto_path = "../assets/images/alat/" . $_POST['nama_alat'] . "." . $ext;
$stampil_foto_path = "assets/images/alat/" . $_POST['nama_alat'] . "." . $ext;
if (!file_exists("../assets/images/alat")) {
    mkdir("../assets/images/alat");
}
move_uploaded_file($foto_path);

$data = [
    $_POST['id'],
    $_POST['id_alat'],
    $_POST['nama_alat'],
    $stampil_foto_path
];
}else{
    $data = [
        $_POST['id'],
        $_POST['id_alat'],
        $_POST['nama_alat'],
        null
    ];
}
$stmt->execute($data);

```

Gambar 3.6 Coding Lokasi Penyimpanan Foto

Selanjutnya membuat 2 variabel *\$foto_path* dan *\$foto_tampil_path* seperti pada gambar 3.6 untuk menggabungkan *value* dari *String* lokasi penyimpanan foto ditambah variabel nama alat yang didapatkan dari *input* yang *user* berikan ditambah format ekstensi foto tersebut. Jika folder tempat penyimpanan foto belum dibuat, maka akan dibuat secara otomatis dan foto akan dipindahkan ke lokasi berdasarkan *value* dari variabel *\$foto_path*.

Proses pemindahan foto akan dilakukan jika *user* mengirimkan foto melalui sistem aplikasi. Jika *user* tidak mengirimkan foto, maka sistem aplikasi hanya mengirimkan *input* dari *user* langsung ke database tanpa melalui proses pemindahan foto.

Pada proses *Read*, hanya membuat *coding* untuk membaca isi dari tabel untuk kemudian ditampilkan ke *user*.

```
require 'koneksi.php';
$pdo = koneksiDb();

$sql = "SELECT * FROM alat_kalibrasi";

$hasil = $pdo->prepare($sql);
$hasil->execute();
-?>
<h1 class="h2 mt-3">Alat Kalibrasi</h1>
<div class="table-responsive-mt-3">
<table class = "table" border="1">
  <tr>
    <th>No. </th>
    <th>No. Inventory</th>
    <th>Nama Alat</th>
    <th>Foto Alat</th>
  </tr>
  <tbody>
    <tr>
      <td>?</td>
      <td>?</td>
      <td>?</td>
      <td>?</td>
    </tr>
  </tbody>
</table>
</div>
```

Gambar 3.7 Coding untuk proses *Read*

Pada gambar 3.7 merupakan *coding* yang dibuat untuk proses *Read* yang dimulai dari membuat koneksi ke database menggunakan *class* PDO yang sudah dijelaskan pada gambar 3.2. selanjutnya membuat variabel *\$sql* yang berisikan permintaan untuk

membaca seluruh isi dari tabel `alat_kalibrasi` untuk kemudian membuat *prepared statement* dan menjalankan variabel `$sql` tersebut. Data yang akan dibaca merupakan informasi alat kalibrasi seperti nomor *inventory*, nama alat, dan foto alat kalibrasi. Data – data tersebut disajikan dalam bentuk tabel dengan sebuah fungsi untuk mengambil data dari database kemudian menyimpan nya ke dalam *Array* dengan variabel `$data`. *Array* adalah sebuah wadah untuk menyimpan data – data dengan format yang sama dalam jumlah tertentu. Pada gambar 3.7 menggunakan perulangan *while* untuk mengambil *entry data* satu per satu untuk kemudian disimpan ke dalam *array* `$data` dan ditampilkan ke dalam tabel dengan variabel `$data['']`. Untuk menampilkan foto pada gambar 3.7 menggunakan fungsi *img src* dimana fungsi ini akan mencari nama file foto yang sesuai dengan value dari kolom foto dari database dan menampilkan foto tersebut ke sistem aplikasi.

Pada proses *Update*, pada dasarnya adalah mengelola isi dari data yang sudah ada baik menambah isi data maupun mengubah isi data tersebut. Untuk sistem aplikasi yang telah dibuat oleh tim *back-end development*, proses *Update* diberikan agar *user* dapat mengganti isi dari data seputar alat kalibrasi yang telah terdaftar di dalam database.


```

$sql = "UPDATE alat_kalibrasi
SET id_alat = ?,
    nama_alat = ?,
    foto_alat = ?
WHERE id = ?";
$stmt = $pdo->prepare($sql);
$stmt->execute([
    $_POST['idbarang'],
    $_POST['nama'],
    $stampil_foto_path,
    $_POST['id']
]);

$ext_boleh = ["jpg", "png", "jpeg"];
if(checkFile($_FILES['foto'], $ext_boleh)){
    $ext = getFileExt($_FILES['foto']);
    $temp = $_FILES['foto']['tmp_name'];
    $foto_path = "../assets/images/user/" . $_POST['nama'] . "." . $ext;
    $stampil_foto_path = "assets/images/user/" . $_POST['nama'] . "." . $ext;
    move_uploaded_file($temp, $foto_path);

    $sql = "UPDATE user SET foto = ? WHERE id = ?";
    $stmt = $pdo->prepare($sql);
    $data = [$stampil_foto_path, $_POST['id']];
    $stmt->execute($data);
}else{
    $data = [null, $_POST['id']];
    $stmt->execute($data);
}

```

Gambar 3.8 Coding untuk proses Update

Proses *Update* pada gambar 3.8 dimulai dengan membuat variabel `$sql` yang berisikan permintaan kepada database untuk mengganti isi dari tabel `alat_kalibrasi` dimana hanya item dengan *id* yang sesuai dapat di ganti informasinya jika diharuskan. Proses *filter* dan pemindahan file foto juga dilakukan proses *Update* jika *user* ingin mengganti foto yang sudah lama dengan yang baru. Jika *user* mengganti foto maka data yang dikirimkan ke database adalah data Nomor Inventory, Nama Alat, dan Foto Alat. Sedangkan, jika tidak mengganti foto maka hanya data Nomor Inventory dan Nama Alat

yang akan dikirimkan ke database untuk menggantikan data yang lama.

Proses *Delete* merupakan proses yang memerlukan *coding* paling sedikit dibandingkan dengan 3 proses CRUD lainnya.

```
$sql = "DELETE FROM user WHERE id = ?";  
$stmt = $pdo->prepare($sql);  
$stmt->execute([$GET['id']]);
```

Gambar 3.9 Proses untuk *Delete*

Proses *Delete* ini diawali dengan membuat variabel `$sql` untuk mengirimkan permintaan kepada database untuk menghapus *entry data* dari tabel berdasarkan *id* yang telah dipilih sebelumnya. Kemudian membuat *prepared statement* dan mengeksekusi variabel `$sql`.

3.3.2 Ikut Kegiatan Kalibrasi Alat Kesehatan dan Input Data Sertifikasi Alat Kesehatan (Minggu ke 7 – 8)

Pada minggu pertama program kerja magang dimulai, Bapak Ali Faturochman menjelaskan tentang kegiatan – kegiatan yang perusahaan lakukan. PT Polaritas Multitrans Technology seringkali menerima *order* untuk melakukan kalibrasi alat rumah sakit di sekitar Kabupaten Tangerang sampai ke Kabupaten Serang, Banten. Pada minggu ke 7 hingga ke 8 ditawarkan untuk ikut kegiatan kalibrasi alat – alat rumah sakit untuk RS Eka Hospital yang terletak

di kota BSD dan membantu dalam melakukan *input data* seputar data – data sertifikasi alat – alat kesehatan rumah sakit.

Kegiatan kalibrasi yang dilakukan pada alat – alat kesehatan rumah sakit dimulai dari menganalisa kemampuan alat kesehatan rumah sakit tersebut untuk mengetahui kondisi alat sebelum kalibrasi dilakukan. Kemudian melihat acuan yang dikeluarkan oleh Komite Akreditasi Nasional yang sesuai untuk alat kesehatan rumah sakit yang akan dikalibrasi. Alat kalibrasi untuk peralatan rumah sakit berbagai macam dan memiliki tata cara yang berbeda pada setiap peralatan rumah sakit agar hasil kalibrasi lebih maksimal. Setelah kalibrasi dilakukan, teknisi melakukan *input data* berupa laporan hasil kalibrasi alat kesehatan rumah sakit untuk diteruskan ke divisi Administrasi dan divisi Akuntansi untuk permintaan biaya kepada rumah sakit yang telah dilakukan kalibrasi.

Kegiatan *input data* seputar data – data sertifikasi untuk alat – alat kesehatan rumah sakit yang telah di kalibrasi melibatkan seluruh staf IT untuk melakukan kegiatan ini. Banyak dari karyawan PT Polaritas Multitrans Technology yang melakukan kecurangan terutama pada laporan hasil kalibrasi alat rumah sakit yang seharusnya tidak layak namun dilaporkan sebagai masi layak untuk dipakai. Serta ditemukan beberapa sertifikasi dari hasil kalibrasi peralatan rumah sakit yang pernah di kalibrasi belum dimasukkan ke dalam database. Untuk melakukan *input data* tersebut,

menggunakan media berupa *laptop* dan koneksi internet dengan *interface* menggunakan kabel RJ45 atau *wi-fi*. Kegiatan *input data* selama program kerja magang berlangsung, menggunakan sebuah perangkat lunak berbasis web untuk melakukan *input data* seputar sertifikasi hasil kalibrasi peralatan rumah sakit hingga hasil dari kalibrasi peralatan rumah sakit tersebut.

3.3.3 User Interface Sistem Inventory (Minggu ke 9 – 10)

Tujuan dari pembuatan sistem *Inventory* adalah untuk memudahkan divisi *Warehouse* dalam memastikan alat kalibrasi yang ada di dalam *warehouse*, serta mengatur koleksi alat kalibrasi yang ada di dalam gudang dikarenakan koleksi alat kalibrasi sudah semakin banyak dan menyusahakan pihak *Warehouse* dalam mengatur gudang penyimpanan. Bapak Ali Faturochman dan divisi *Warehouse* memberikan masukan agar *user interface* yang dibuat mengutamakan kemudahan dalam penggunaan dibandingkan keindahan tampilan dikarenakan *user* yang menggunakan *user interface* ini hanyalah karyawan dari divisi *warehouse*.



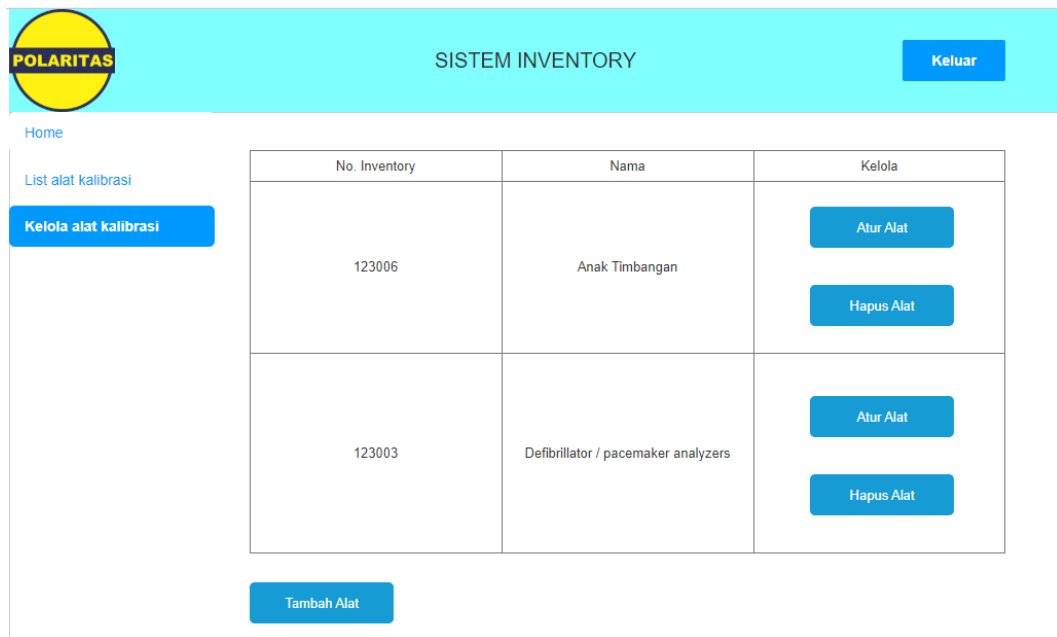
Gambar 3.10 Halaman Utama

Pada gambar 3.1 adalah halaman utama dari *user interface* sistem inventory. Dalam halaman utama ini user tidak diminta untuk memasukkan *username* dan *password* dikarenakan komputer yang akan digunakan sudah menerapkan sistem *fingerprint scanner* untuk memastikan hanya pihak *warehouse* yang dapat mengakses sistem *Inventory* ini. Pintu masuk ke dalam ruangan komputer di dalam *Warehouse* sudah dilengkapi dengan sistem keamanan CCTV yang beroperasi 24 jam untuk memastikan tidak ada pihak yang dapat meretas masuk ke dalam komputer divisi *Warehouse* dan merubah isi database dan merugikan perusahaan dalam jumlah yang besar, terlebih merugikan *Business Value* sehingga akan mempersulit PT Polaritas Multitrans Technology dalam mencari klien baru.



Gambar 3.11 Halaman List Alat Kalibrasi

Pada gambar 3.2 adalah halaman list alat kalibrasi. Disini divisi *Warehouse* dapat melihat alat kalibrasi apa saja yang tersimpan di *warehouse* beserta foto dari alat kalibrasi tersebut. Informasi alat kalibrasi tersebut disajikan dalam bentuk tabel dengan judul kolom *Nomor Inventory* yang mengambil data dari *database* perusahaan dengan tabel *alat_kalibrasi* yang dipilih. Dalam tahap pembuatan *user interface*, Bapak Ali Faturochman memberikan data *dummy* atau data palsu untuk mengisi tabel agar dapat memperkirakan tabel yang telah dibuat cukup untuk halaman ini atau tidak. Sambil Bapak Ali Faturochman memberikan data *dummy*, beliau juga menjelaskan fungsi dari alat – alat kalibrasi tersebut. Pada gambar 3.11 ada sebuah tabel yang berisikan informasi tentang alat kalibrasi yang telah terdaftar dalam database milik PT Polaritas Multitrans Technology.



Gambar 3.12 Halaman Kelola Alat Kalibrasi

Pada gambar 3.3 adalah halaman untuk mengelola informasi alat – alat kalibrasi. Di halaman ini *user* diperlihatkan data – data alat kalibrasi yang diambil dari database. Tetapi untuk halaman ini, *user* hanya diperlihatkan data berupa *id* alat kalibrasi beserta namanya. Kolom foto diganti dengan 2 tombol pada setiap alat untuk mengubah informasi tentang alat kalibrasi dan menghapus *entry* data alat kalibrasi. Berikut penjelasan detail dari setiap tombol pada halaman ini:

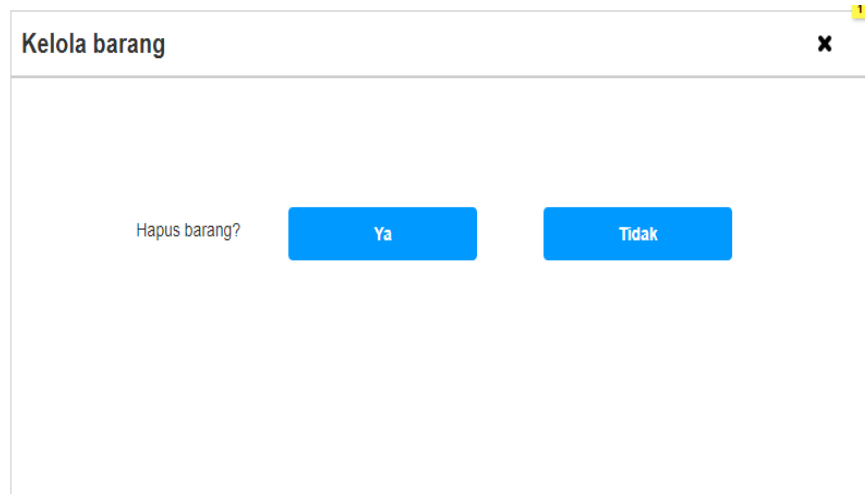
1. Atur alat. Disini *user* dapat mengubah deskripsi barang dari nama alat kalibrasi dan foto alat kalibrasi tersebut.

The image shows a web-based form titled "Tambah barang" (Add item) with a close button in the top right corner. The form contains three input fields: "ID alat", "Nama alat", and "Foto". The "Foto" field has a "Browse" button next to it. At the bottom of the form is a blue "Simpan" (Save) button.

Gambar 3.4 Pop Up Tambah Alat

Gambar 3.4 akan muncul berupa *pop-up* ketika user memilih tombol Tambah Alat. *User* dapat memasukkan informasi seputar *id* alat kalibrasi yang didapatkan dari sistem penomoran untuk pendaftaran alat kalibrasi, nama alat, dan foto dari alat kalibrasi tersebut. Setelah semua data dimasukkan *user* dapat memilih tombol Simpan untuk finalisasi input data alat kalibrasi.

1. Hapus alat. Disini user dapat menghapus *entry* alat kalibrasi yang tersimpan di dalam *database*. Pilihan ini disediakan karena ada beberapa alat kalibrasi yang memiliki masa pakai. Ketika masa pakai tersebut habis, alat kalibrasi tersebut harus dibuang atau dikirim ke tempat peremajaan alat kalibrasi.



Gambar 3.5 Pop Up Konfirmasi Hapus Alat

Setelah *user* memilih untuk menghapus alat dari sistem aplikasi, akan muncul *pop-up* seperti pada gambar 3.5. *user* akan diminta konfirmasi untuk menghapus alat dari sistem atau tidak dikarenakan alat kalibrasi yang sudah habis masa pakainya memiliki kemungkinan untuk dipakai Kembali setelah dilakukan peremajaan pada alat tersebut.

2. Tambah alat. Pilihan ini disediakan untuk mendaftarkan alat kalibrasi yang baru saja diakuisisi oleh perusahaan ke dalam *database*.

The image shows a web-based form titled "Tambah Alat" (Add Tool). The form is contained within a window with a title bar that includes the text "Tambah Alat" and a close button (X) in the top right corner. The form itself has a white background and a thin border. It contains three input fields arranged vertically: "ID alat", "Nama alat", and "Foto". The "Foto" field is accompanied by a "Browse" button. At the bottom center of the form is a prominent blue button labeled "Simpan" (Save).

Gambar 3.6 Pop Up Tambah Alat

Setelah *user* memilih tombol Tambah Alat, akan muncul *pop-up* seperti pada gambar 3.6. Di sini *user* dapat meng-input data alat kalibrasi baru berupa *id* alat yang diambil dari sistem penomoran alat – alat kalibrasi yang terdaftar di *Warehouse*, nama alat kalibrasi, dan foto dari alat kalibrasi tersebut. Setelah semua data berhasil dimasukkan, *user* dapat melakukan finalisasi dengan memilih tombol Simpan dan akan langsung kembali ke halaman Kelola Alat Kalibrasi. Jika *user* tidak jadi menambah alat kalibrasi baru, *user* tersebut dapat memilih *icon X* yang terletak di ujung kanan atas pada tampilan *pop-up* display

3.4 Kendala yang Dihadapi

Selama melakukan pelaksanaan pekerjaan magang ada beberapa kendala – kendala yang dihadapi antara lain sebagai berikut:

1. Perbedaan versi *software* yang digunakan selama program kerja magang berlangsung dan yang dipelajari di masa perkuliahan. Pada *software* Axure Rapid Prototyping pada program kerja magang berlangsung menggunakan Axure versi 9, sedangkan pada masa perkuliahan diajarkan pada Axure versi 6.
2. Adanya rasa sungkan dan malu untuk menyampaikan pendapat karena terdapat perbedaan budaya dan perbedaan pengalaman.
3. Pada saat program magang berlangsung terjadi pandemik virus COVID-19 sehingga pada minggu ke 7 hingga minggu ke 10 PT Polaritas Multitrans Technology menerapkan aturan *work-from-home*, dimana sudah ada rencana untuk ikut kegiatan perusahaan mengkalibrasi alat kesehatan rumah sakit di RS Eka Hospital Tangerang harus dibatalkan demi keselamatan para karyawan termasuk mahasiswa/I yang sedang menjalankan program magang.

3.5 Solusi atas Kendala

Untuk mengatasi kendala yang dialami selama program kerja magang berlangsung, berikut merupakan solusi atas kendala yang terjadi selama program magang:

1. Mempelajari fitur dan cara menggunakan *software* Axure Rapid Prototyping versi 9 melalui *internet* dan media *youtube* serta melihat forum seputar membuat *user interface* menggunakan *software* Axure versi 9.
2. Memberanikan diri untuk menyampaikan pendapat dengan cara yang sopan santun dan belajar untuk beradaptasi serta membaaur dengan lingkungan kerja yang terdiri dari berbagai macam budaya dan tantangan.
3. Memberikan saran untuk menggunakan *software* yang memungkinkan interaksi tatap muka dengan lebih dari 2 orang seperti *software* Zoom, Google Hangouts Meet, dan Microsoft Teams.