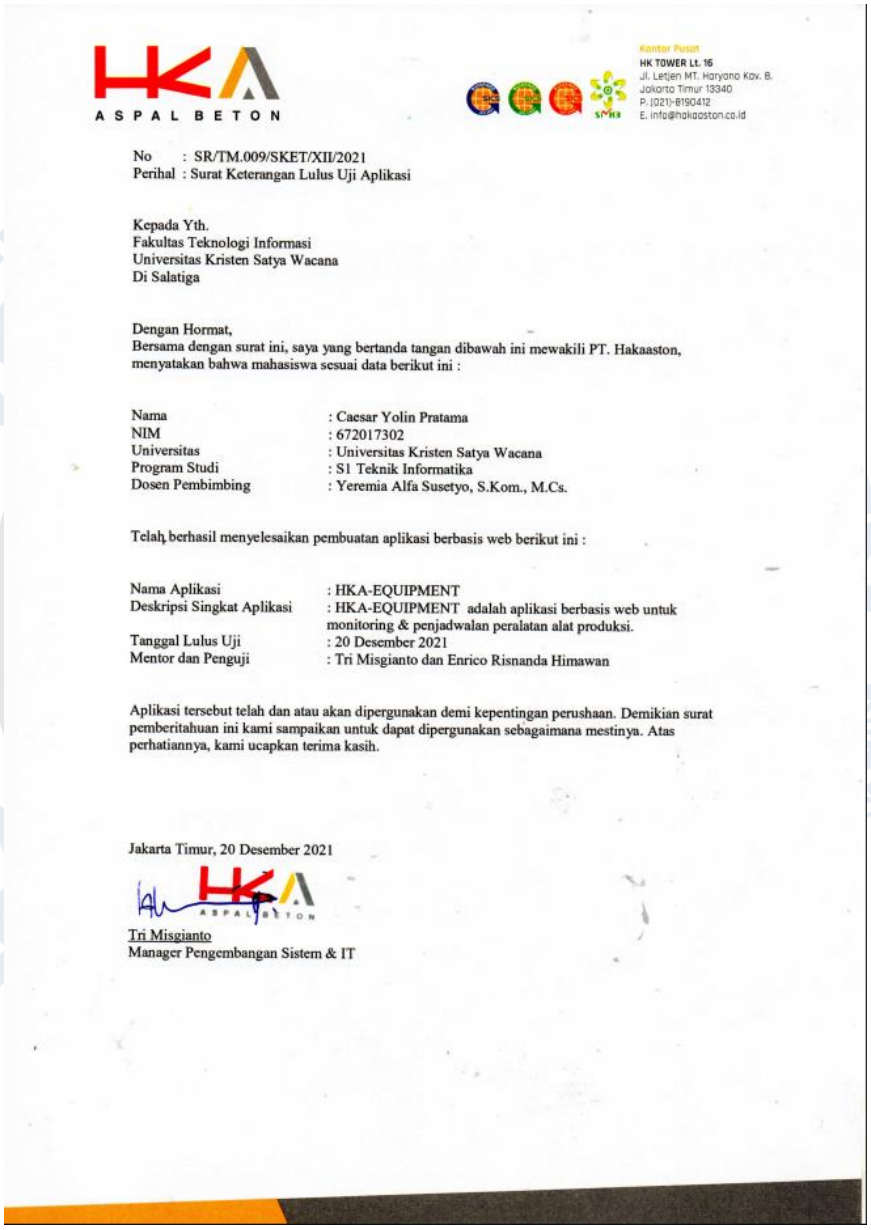
Lampiran

Surat Keterangan Lulus Uji



HKA ASPAL BETON

Kantor Pusat
HK TOWER Lt. 16
Jl. Letjen MT. Haryono Kav. B.
Jakarta Timur 13340
P. (021)-8190412
E. info@hakaaston.co.id

No : SR/TM.009/SKET/XII/2021
Perihal : Surat Keterangan Lulus Uji Aplikasi

Kepada Yth.
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Di Salatiga

Dengan Hormat,
Bersama dengan surat ini, saya yang bertanda tangan dibawah ini mewakili PT. Hakaaston, menyatakan bahwa mahasiswa sesuai data berikut ini :

Nama	: Caesar Yolin Pratama
NIM	: 672017302
Universitas	: Universitas Kristen Satya Wacana
Program Studi	: S1 Teknik Informatika
Dosen Pembimbing	: Yeremia Alfa Susetyo, S.Kom., M.Cs.

Telah berhasil menyelesaikan pembuatan aplikasi berbasis web berikut ini :

Nama Aplikasi	: HKA-EQUIPMENT
Deskripsi Singkat Aplikasi	: HKA-EQUIPMENT adalah aplikasi berbasis web untuk monitoring & penjadwalan peralatan alat produksi.
Tanggal Lulus Uji	: 20 Desember 2021
Mentor dan Penguji	: Tri Misgianto dan Enrico Risnanda Himawan

Aplikasi tersebut telah dan atau akan dipergunakan demi kepentingan perusahaan. Demikian surat pemberitahuan ini kami sampaikan untuk dapat dipergunakan sebagaimana mestinya. Atas perhatiannya, kami ucapkan terima kasih.

Jakarta Timur, 20 Desember 2021

HKA ASPAL BETON

Tri Misgianto
Manager Pengembangan Sistem & IT

Source Code

Kode Program 1 Fungsi Tabel Data Alat

```
1   public function data_mesin(){
2       $alat = DB::table('alat')
```

```
3        ->where('jam', '>=', 1000)
4        ->where('repair', '=', 'belum')
5        ->get();
6
7
8        $user = Auth::user();
9        $data= DB::table('alat')
10        ->join('posisi', 'alat.posisi', '=', 'posisi.id')
11        ->where('posisi','=', $user->posisi)
12        ->select('alat.*', 'posisi.namaPosisi')
13        ->get();
14
15
16
17        return view('adminpu.v_adminpu_data_mesin',
          ['data' =>   $data, 'alat' => $alat]);}
```

Pada kode program 1 ditampilkan sebuah fungsi data_alat() yang terdapat pada AdminUnitProduksiController. Fungsi ini berfungsi sebagai pengirim data yang akan ditampilkan pada *view*. Pada baris 2-5 dibuat sebuah *variable* $alat, yang mana pada *variable* tersebut diisikan kumpulan beberapa objek yang diambil dari model 'alat'. *Value* dalam *variable* ini nantinya akan berfungsi sebagai pembuatan notifikasi pada *view*. Pada baris 8 dibuat lagi sebuah *variable* '$user', yang mana *variable* ini memiliki *value* berupa data dari hasil authentikasi *login user* (data – data *user* yang sedang *login* kedalam sistem). Pada baris 9 dibuat lagi sebuah *variable* dengan memasukkan data dari alat yang akan ditampilkan pada tabel dibagian *view*. Pada Baris 17 'return view' berarti mengembalikan fungsi tersebut ke sebuah *view* bernama 'v_adminpu_data_mesin' yang ada pada file view atau admin pusat dengan mengirimkan semua data yang ada pada *variable* yang telah dibuat sebelumnya.

Kode Program 2 *View* Tabel Data Alat

```
1 @foreach($data as $u)
2
3                          <tr>
4                           <td>{{ ++$no  }}</td>
5                           <td>{{ $u->nama_alat }}</td>
6                           <td>{{ $u->merk }}</td>
7                           <td>{{ $u->tahun }}</td>
8                           <td>{{ $u->umur_alat }}</td>
9                           <td>{{ $u->namaPosisi }}</td>
10                          <td>{{ $u->kondisi }}</td>
11                          <td>{{ $u->status }}</td>
```

```
12                          <td>{{ $u->jam }}</td>
```

Dalam pembuatan sebuah aplikasi sebelum digunakan oleh publik harus melewati tahap pengujian sistem. Pengujian sistem bertujuan mengetahui apakah aplikasi dapat berjalan sesuai fungsinya atau tidak serta menemukan celah pada aplikasi. Berikut merupakan pengujian sistem *Black Box Testing* pada aplikasi HKA-EQUIPMENT yang dapat dijelaskan dalam bentuk tabel di bawah ini. Lalu pada *view* nantinya *variable* yang telah dikirimkan melalui *controller* akan diproses. Seperti contohnya pada baris 1, @foreach($data as $u), artinya *variable* $data yang telah dibuat pada *controller* akan dilakukan *looping* untuk menampilkan data yang ada didalamnya. Pada line 4-12 adalah *syntax* yang akan menampilkan data pada *variable* $data. Hasilnya adalah halaman data alat pada admin unit produksi.

Kode Program 3 *Form* Tambah Jam Alat

```
1 </table>
2                <div  class="modal"  id="modal-jam"  tabindex="-1"
role="dialog"       aria-labelledby="exampleModalLabel"       aria-
hidden="true">
3                 <div  class="modal-dialog  modal-dialog-centered"
role="document">
4            <div class="modal-content">
5                <div class="modal-header">
6                                          <h5  class="modal-title"
id="exampleModalLabel">Form Input Jam Kerja</h5>
7                       <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
8                        <span aria-hidden="true">&times;</span>
9                     </button>
10                  </div>
11                  <div class="modal-body">
12                    <form id="jamForm" action="" method="post"
>
13                          {{ csrf_field() }}
14                          <div class="mb-3">
15                                            <label  for="regionDesc"
class="form-label">Jam Kerja</label>
16                                  <input type="number" class="form-
control"  name="jam"  id="regionDesc"  placeholder="Masukkan  Jam
Kerja" required>
17                          </div>
18
19                            <button type="submit" class="btn btn-
primary btn-block">Simpan</button>
20                      </form>
```

```
21                  </div>
22
23                  <div class="modal-footer">
24                          <button type="button" class="btn btn-
secondary btn-block" data-dismiss="modal">Batal</button>
25                  </div>
```

Pada kode program 3 adalah *file view* yang menampikan *form* tambah jam alat. Dari *view* tersebut nantinya akan diambil *value* dari inputan dengan nama jam. *Value* yang didapat akan digunakan pada *controller.*

Kode Program 4 Fungsi Tambah Jam

```
1 public function tambah_jam(Request $request, $id){
2       $this->validate($request,[
3           'jam' => 'required|numeric',
4
5
6       ],
7       $message = [
8    'jam.required' => 'Kolom jam kerja harus diisi',
9    'jam.numeric' => 'Kolom jam kerja harus berupa angka',
10       ]);
11    $data = DB::table('alat')->where('id', $id)->first();
12       $jam = $data->jam + $request->jam;
13
14       if($data->repair == 'sudah'){
15           DB::table('alat')
16               ->where('id', $id)
17               ->update([
18                   'jam' => $jam,
19                   'repair' => 'belum'
20               ]);
21       }
22       else {
23           DB::table('alat')
24               ->where('id', $id)
25               ->update([
26                   'jam' => $jam,
27               ]);
28       }
```

Pada kode program 4 ditampilkan sebuah fungsi tambah_jam() yang berfungsi menambahkan jam yang akan diinputkan *user* ke dalam *database* alat atribut jam. Pada baris 1-10, dibuat suatu *validation* yang akan mengatur jenis data dan juga validasi lainnya pada *form* yang dibuat pada *view*. Pada baris 11 dibuat sebuah *variable* $data

yang akan menyimpan data dari ‘alat’ yang memiliki ‘id’ sama dengan $id yang didapatkan dari tabel alat. Lalu line setelahnya dibuat lagi *variable* $jam yang berisi ‘jam’ dari alat yang telah didapatkan pada line sebelumnya, dan akan dijumlahkan dengan $request yang didapatkan dari *form* pada *view*. Setelahnya dibuat sebuah kondisi, dimana jika atribut *repair* pada alat tersebut menunjukkan ‘sudah’ maka akan dilakukan *update* pada atribut ‘jam’ dan juga ‘repair’. Atribut ‘jam’ diisi dengan *variable* yang $jam sebelumnya dan ‘repair’ diisi dengan ‘belum’. Jika tidak maka hanya akan memperbarui atribut ‘jam’ saja.