

# PEREKAMAN KEHADIRAN DARING OTOMATIS

REINALTA SUGIANTO—2017730035

## 1 Data Skripsi

Pembimbing utama/tunggal: **Pascal Alfadian Nugroho**

Pembimbing pendamping: -

Kode Topik : **PAN5191**

Topik ini sudah dikerjakan selama : **1 semester**

Pengambilan pertama kali topik ini pada : Semester **51 - Ganjil 21/22**

Pengambilan pertama kali topik ini di kuliah : **Skripsi 1**

Tipe Laporan : **B** - Dokumen untuk reviewer pada presentasi dan **review Skripsi 1**

## 2 Latar Belakang

Perkuliahan di UNPAR biasanya membutuhkan perekaman kehadiran untuk mengetahui kehadiran mahasiswa dan dosen, bagi mahasiswa UNPAR perekaman kehadiran biasanya dilakukan dengan melakukan tanda tangan pada daftar kehadiran atau dicatat langsung oleh dosen yang memanggil mahasiswanya, sedangkan bagi dosen UNPAR perekaman kehadiran dilakukan dengan menggunakan *fingerprint*. Perekaman kehadiran diperkirakan membutuhkan waktu sekitar kurang dari 5 detik.

Pada tahun 2020 terjadi pandemi Covid-19 di seluruh negara. Pandemi Covid-19 masuk ke Indonesia pada awal bulan Maret tahun 2020. Covid-19 adalah penyakit yang disebabkan oleh virus *severe acute respiratory syndrome coronavirus 2* (SARS-CoV-2). Penularan virus Covid-19 terjadi saat seseorang menyentuh barang yang sudah terkontaminasi oleh droplet orang yang terkena virus Covid-19 atau terkena droplet orang lain saat berinteraksi langsung dengan orang yang terkena virus Covid-19. Akibat pandemi Covid-19 yang dapat menular ini, maka hampir seluruh kegiatan di Indonesia dilakukan secara daring untuk mengurangi interaksi orang secara langsung yang dapat meningkatkan angka penularan virus tersebut.

Pembelajaran secara daring diberlakukan oleh UNPAR di akhir bulan Maret untuk seluruh kegiatan perkuliahan demi mencegah penularan virus Covid-19. Akibat diberlakukannya pembelajaran secara daring, maka perekaman kehadiran di UNPAR dilakukan dengan menggunakan aplikasi atau situs web milik UNPAR. Cara perekaman kehadiran secara daring di UNPAR ini membutuhkan waktu lebih agar dapat tercatat perekaman kehadirannya, karena butuh waktu untuk membuka situs web serta perlu memasukkan *email* dan *password* hingga akhirnya melakukan perekaman kehadiran.

Selenium adalah *open-source framework* pengujian otomatisasi untuk aplikasi web. WebDriver menggunakan API otomatisasi *browser* yang disediakan oleh vendor *browser* untuk mengontrol *browser* dan melakukan pengujian. API WebDriver ini seolah-olah membuat pengguna secara langsung mengoperasikan *browser*, padahal dijalankan secara otomatis langsung oleh API WebDriver tersebut. Selenium WebDriver adalah sebuah *tools* yang berguna untuk melakukan otomatisasi terhadap web pada *browser*. Selenium WebDriver ini tersedia untuk bahasa pemrograman Ruby, Java, Python, C#, dan JavaScript. Pembuatan Perekaman kehadiran daring otomatis ini akan menggunakan Selenium WebDriver dengan bahasa pemrograman Python.

Pada skripsi ini, akan dibuat sebuah perangkat lunak yang dapat melakukan perekaman kehadiran otomatis dengan sistem menerima rangsangan satu “klik” sehingga dapat melakukan hal-hal berikut :

1. Membuka peramban.
2. Membuka situs web perekaman kehadiran.
3. Mengisi dan *login* dengan *username* serta *password* yang diambil dari file konfigurasi.

4. Melakukan rekam kehadiran.

perangkat lunak ini bertujuan agar mahasiswa dan dosen dapat melakukan perekaman kehadiran secara online dengan lebih mudah serta mengurangi waktu yang dibutuhkan untuk berinteraksi dengan aplikasi atau situs web dan bukan untuk mempercepat waktu agar kehadiran terekam, sehingga membuat waktu perekaman kehadiran secara daring dapat menyamai waktu perekaman kehadiran secara luring.

### 3 Rumusan Masalah

Rumusan masalah yang akan dibahas di skripsi ini adalah sebagai berikut :

- Bagaimana cara membangun program Perekaman Kehadiran Daring Otomatis?
- Bagaimana cara mengurangi waktu interaksi dengan aplikasi atau situs web untuk merekam kehadiran?

### 4 Tujuan

Tujuan yang ingin dicapai dari penulisan skripsi ini sebagai berikut :

- Membangun program menggunakan Selenium WebDriver.
- Membuat program yang mampu menerima rangsangan satu tombol untuk melakukan beberapa hal menggunakan Selenium.

### 5 Detail Perkembangan Pengerjaan Skripsi

Detail bagian pekerjaan skripsi sesuai dengan rencan kerja/laporan perkembangan terkahir :

#### 1. Melakukan studi literatur mengenai Selenium WebDriver.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Studi literatur mengenai Selenium WebDriver dilakukan dengan cara membaca situs web resmi Selenium Webdriver, situs-situs pembelajaran terkait, dan menonton video pembelajaran dari youtube. Selenium adalah *open-source framework* pengujian otomatisasi untuk aplikasi web. Selenium ini adalah WebDriver yang merupakan sebuah *interface* untuk menulis suatu instruksi yang dapat dijalankan secara otomatis dan bergantung pada *browser*. Selenium WebDriver adalah sebuah *tools* yang berguna untuk melakukan otomatisasi terhadap web pada *browser*. Selenium WebDriver menggunakan API otomatisasi *browser* yang disediakan oleh vendor *browser* untuk mengontrol *browser* dan melakukan pengujian. API WebDriver ini seolah-olah membuat pengguna secara langsung mengoperasikan *browser*, padahal dijalankan secara otomatis langsung oleh *API* WebDriver tersebut. Selenium WebDriver ini tersedia untuk bahasa pemrograman Ruby, Java, Python, C#, dan JavaScript. Selenium WebDriver memiliki berbagai fungsi, yaitu Navigasi *Browser* serta Menemukan elemen. Salah satu teknik mendasar untuk dipelajari saat menggunakan WebDriver adalah cara menemukan elemen di halaman web. WebDriver menyediakan berbagai cara untuk menemukan elemen, terdapat delapan strategi menemukan lokasi elemen yang berbeda di WebDriver:

- (a) *Id* : Menemukan elemen yang atribut ID-nya cocok dengan nilai pencarian.
- (b) *Class name*: Menemukan elemen yang nama kelasnya berisi nilai pencarian.
- (c) *CSS selector*: Menemukan elemen yang cocok dengan pemilihan *Cascading Style Sheets* (CSS). Pemilihan pada CSS adalah pola yang digunakan untuk memilih elemen dengan *style* yang diinginkan.

- (d) *Name*: Menemukan elemen yang atribut *name* yang cocok dengan nilai pencarian.
- (e) *Link text*: Menemukan elemen *link* yang teksnya terlihat cocok dengan nilai pencarian.
- (f) *Partial link text*: Menemukan elemen *link* yang teksnya terlihat berisi nilai pencarian. Jika beberapa elemen cocok, hanya yang pertama yang akan dipilih.
- (g) *Tag name*: Menemukan elemen yang nama tagnya cocok dengan nilai pencarian.
- (h) *XPath*: Menemukan elemen yang cocok dengan ekspresi *XML Path Language* (XPath).

WebDriver secara umum dapat dikatakan memiliki API pemblokiran, karena ini adalah *library* di luar proses yang menginstruksikan browser apa yang harus dilakukan dan karena platform web secara intrinsik memiliki sifat asinkron atau tidak dilakukan secara *real time*, maka WebDriver tidak melacak status *Document Object Model* (DOM) yang aktif dan *real time*, sehingga selenium memiliki fungsi *waits* untuk mengatasi hal tersebut. Berikut ini adalah beberapa fungsi *Waits* pada selenium.

- (a) *Implicit wait*: memberi tahu WebDriver untuk melakukan polling DOM selama jangka waktu tertentu ketika mencoba menemukan elemen. Pengaturan awalnya adalah 0 detik, artinya dinonaktifkan. Setelah disetel, maka *wait implicit* disetel untuk masa pakai sesi yang sudah disetel.
- (b) *Explicit wait*: mengizinkan kode untuk menghentikan eksekusi program, atau membekukan *thread*, hingga suatu kondisi dapat teratasi. Kondisi ini dipanggil dengan frekuensi tertentu sampai batas waktu tunggu terlewati.

Selenium Webdriver sudah didokumentasikan pada Bab 2 Dasar Teori.

## 2. Mempelajari bahasa pemrograman python.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Bahasa pemrograman dengan python dipelajari secara mandiri dengan cara membaca situs-situs pembelajaran terkait, menonton video pembelajaran dari youtube, dan mengambil mata kuliah Pengantar Penambangan Data dengan Python yang melakukan praktek dan eksplorasi menggunakan bahasa pemrograman python.

## 3. Mempelajari cara menggunakan Selenium.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Mempelajari cara menggunakan selenium dengan studi literatur tentang selenium webdriver terlebih dahulu, lalu melakukan eksplorasi menggunakan selenium tersebut secara langsung menggunakan bahasa pemrograman python terhadap situs web Portal Mahasiswa UNPAR. Berikut ini merupakan kode program hasil dari ekplorasi percobaan menggunakan selenium.

Listing 1: Eksplorasi kode Program menggunakan Selenium

```

1  import config_file as cfg
2  from selenium import webdriver
3  from selenium.webdriver.common.by import By
4  from selenium.webdriver.support import expected_conditions as EC
5  from selenium.webdriver.support.ui import WebDriverWait
6  driver = webdriver.Chrome(executable_path='D:\Skripsi\python\chromedriver.exe')
7  driver.maximize_window()
8  portal = driver.get("https://studentportal.unpar.ac.id/")
9  btnIn = driver.find_element(By.CSS_SELECTOR, "#login-button").click()
10 inputEmail= driver.find_element(By.CSS_SELECTOR, "#username").send_keys(cfg.
    drivers_config["email"])
11 btnNext = driver.find_element(By.CSS_SELECTOR, "#next_login").click()
12 inputPassword = WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.
    CSS_SELECTOR, "#password")))
13 inputPassword.click()
14 inputPassword.send_keys(cfg.drivers_config["pass"])

```

```

15     btnLogin = driver.find_element(By.CSS_SELECTOR, "#appPass > div.login__form > button")
        .click()

```

Pada contoh eksplorasi kode program di atas akan dijelaskan perbaris sebagai berikut:

- (a) Baris 1: melakukan import file konfigurasi yang telah dibuat.
- (b) Baris 2: melakukan import webdriver dari selenium yang berguna untuk membuka *browser*.
- (c) Baris 3: melakukan import By untuk menemukan elemen yang ingin dicari.
- (d) Baris 4: melakukan import expected\_conditions yang berguna mengharapkan suatu kondisi dapat terjadi.
- (e) Baris 5: melakukan import WebDriverWait berguna untuk fungsi *Waits* pada selenium untuk memberi waktu melakukan suatu perintah.
- (f) Baris 6: memanggil webdriver yang ingin digunakan, yaitu Google Chrome dan diisi letak file chromedriver.exe disimpan.
- (g) Baris 7: untuk memberi perintah *fullscreen* pada *browser* Google Chrome.
- (h) Baris 8: *String* portal menggunakan *method get* yang berguna untuk menuju situs web Portal Mahasiswa UNPAR.
- (i) Baris 9: *String* btnIn untuk menemukan elemen dari sebuah tombol menggunakan *CSS selector* dengan nama "#login-button", lalu menggunakan *method click* untuk menekan tombol tersebut secara otomatis.
- (j) Baris 10: *String* inputEmail untuk menemukan elemen dari sebuah *input teks* menggunakan *CSS selector* dengan nama "#username", lalu menggunakan *method send\_keys* untuk memasukan email mahasiswa yang sudah dibuat di file konfigurasi untuk dipanggil.
- (k) Baris 11: *String* btnNext untuk menemukan elemen dari sebuah tombol menggunakan *CSS selector* dengan nama "#next\_login", lalu menggunakan *method click* untuk menekan tombol tersebut secara otomatis.
- (l) Baris 12: *String* inputPassword memanggil fungsi *waits* selama 10 detik untuk menunggu elemen yang dicari berdasarkan *CSS selector* dengan nama "#password" ini dapat ditekan.
- (m) Baris 13: Memerintahkan untuk menekan elemen yang sudah dicari pada baris 12.
- (n) Baris 14: Memerintahkan untuk memasukan *password* yang sudah dibuat di file konfigurasi untuk dipanggil.
- (o) Baris 15: *String* btnLogin untuk menemukan elemen dari sebuah tombol *login* menggunakan *CSS selector*, lalu menekannya.

#### 4. Melakukan survei ke mahasiswa dan dosen Teknik Informatika Universitas Katolik Parahyangan untuk mengetahui berapa lama waktu yang dibutuhkan untuk melakukan absensi secara daring dan luring.

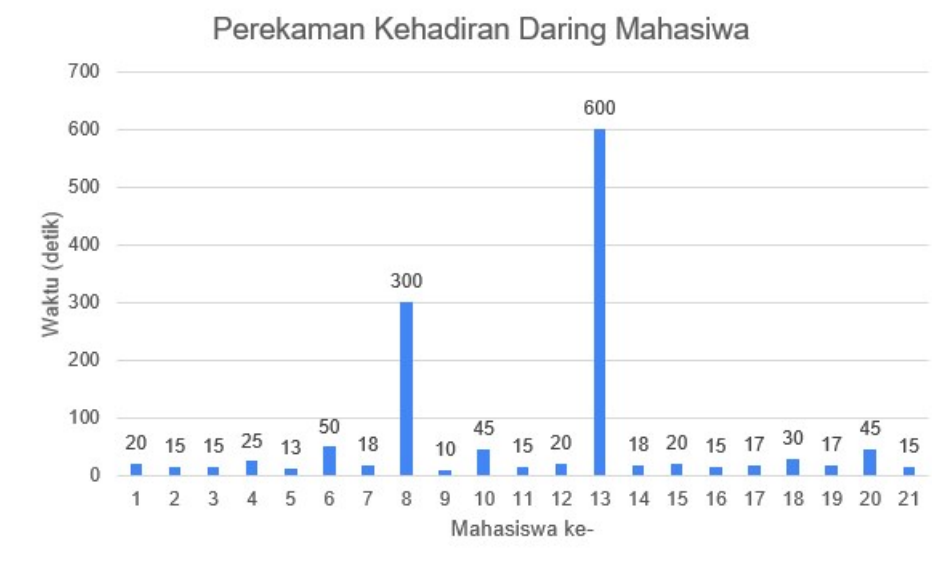
**Status :** Baru ditambahkan pada semester ini.

**Hasil :** Hasil survei telah ditulis dan dianalisis pada Bab 3. Survei dilakukan dengan menyebarkan *Google Form* kepada mahasiswa dan dosen Teknik Informatika Universitas Katolik Parahyangan. Survei ini bertujuan untuk mengetahui waktu yang dibutuhkan untuk melakukan absensi secara daring dan luring.

### 5.1 Hasil Survei Mahasiswa

Berdasarkan hasil survei yang telah diterima dari 21 orang responden yang merupakan mahasiswa Teknik Informatika Universitas Katolik Parahyangan yang terdiri dari mahasiswa angkatan 2017 sampai 2019, dengan pertanyaan yang diajukan kepada responden dan rangkuman jawaban hasil survei sebagai berikut:

- (a) Berapa detik perkiraan waktu interaksi yang Anda butuhkan untuk melakukan perekaman kehadiran daring di <https://studentportal.unpar.ac.id/>, mulai dari membuka *browser*, lalu masuk ke <https://studentportal.unpar.ac.id/>, lalu mengklik tombol presensi?



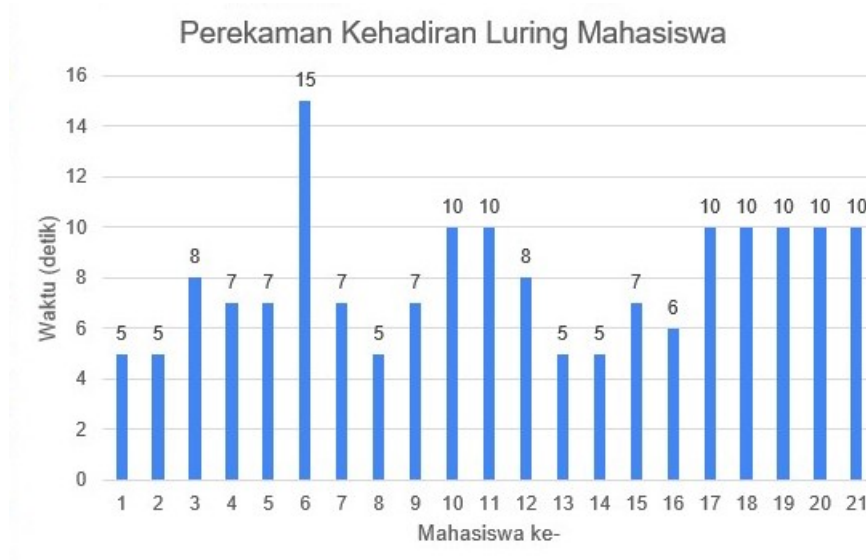
Gambar 1: Histogram Waktu Perekaman Kehadiran Daring Mahasiswa

Pada Gambar 1 merupakan visualisasi dari hasil survei mengenai lama waktu yang dibutuhkan dari 21 mahasiswa untuk melakukan perekaman kehadiran secara daring. Jawaban dari 21 orang responden adalah mulai dari waktu paling cepat 10 detik hingga waktu paling lama 600 detik.

Jumlah Responden	Waktu Perekaman Kehadiran Daring
1 orang	10 detik
1 orang	13 detik
5 orang	15 detik
2 orang	17 detik
2 orang	18 detik
3 orang	20 detik
1 orang	25 detik
1 orang	30 detik
2 orang	45 detik
1 orang	50 detik
1 orang	300 detik
1 orang	600 detik

Jika dihitung rata-rata waktu yang dibutuhkan untuk melakukan perekaman kehadiran daring bagi para mahasiswa adalah 63 detik.

- (b) Berapa detik perkiraan waktu interaksi yang Anda butuhkan untuk melakukan perekaman kehadiran luring menggunakan metode tanda tangan seperti pembelajaran di kelas, mulai dari mengambil kertas absen, lalu tanda tangan, lalu memberikannya ke rekan di sebelah anda?



Gambar 2: Histogram Waktu Perekaman Kehadiran Daring Mahasiswa

Pada Gambar 2 merupakan visualisasi dari hasil survei mengenai lama waktu yang dibutuhkan dari 21 mahasiswa untuk melakukan perekaman kehadiran secara luring. Jawaban dari 21 orang responden adalah mulai dari waktu paling cepat 5 detik hingga waktu paling lama 15 detik.

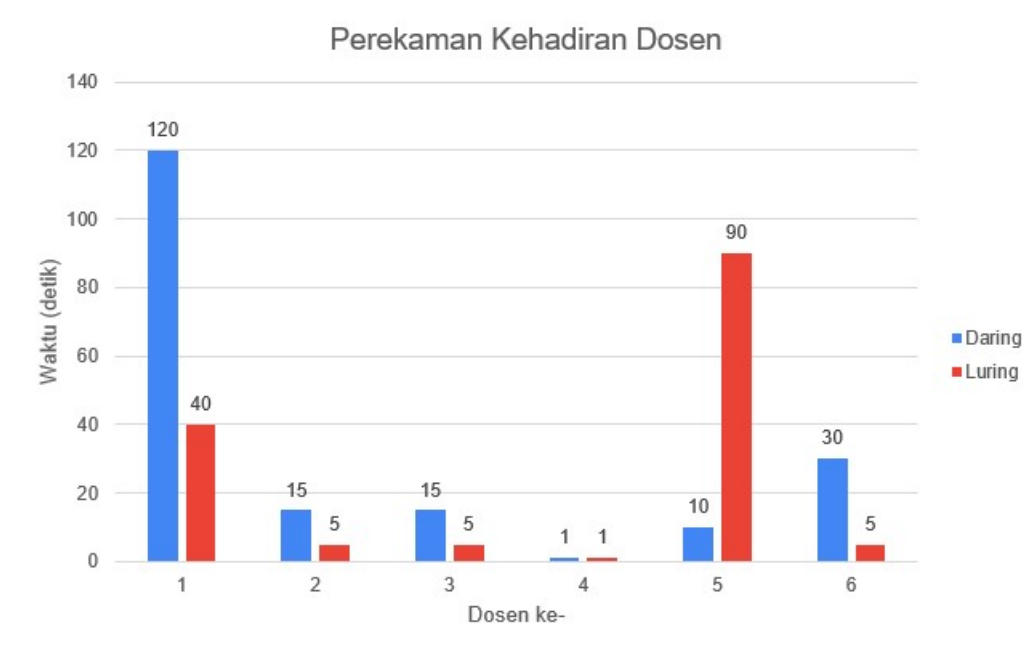
Jumlah Responden	Waktu Perekaman Kehadiran Luring
5 orang	5 detik
1 orang	6 detik
5 orang	7 detik
2 orang	8 detik
7 orang	10 detik
1 orang	15 detik

Jika dihitung rata-rata waktu yang dibutuhkan untuk melakukan perekaman kehadiran luring bagi para mahasiswa adalah 7,95 detik.

Kesimpulan dari hasil survei mahasiswa menunjukkan bahwa rata-rata waktu yang dibutuhkan secara luring adalah 7,95 detik lebih cepat dibandingkan dengan rata-rata waktu yang dibutuhkan secara daring adalah waktu 63 detik

## 5.2 Hasil Survei Dosen

Berdasarkan hasil survei yang telah diterima dari 6 orang responden yang merupakan dosen Teknik Informatika Universitas Katolik Parahyangan, dengan pertanyaan yang diajukan kepada responden dan rangkuman jawaban hasil survei sebagai berikut:



Gambar 3: Histogram Perbandingan Waktu Perekaman Kehadiran Dosen

Pada Gambar 3 merupakan visualisasi dari perbandingan waktu yang dibutuhkan dari 6 dosen Teknik Informatika Universitas Katolik Parahyangan. Grafik warna biru menjelaskan perekaman kehadiran secara daring dan warna merah menjelaskan perekaman kehadiran secara luring. Mayoritas menyatakan bahwa perekaman kehadiran luring lebih cepat dibandingkan daring, hanya dosen ke-4 yang menyatakan waktunya sama dan dosen ke-5 yang menyatakan waktu perekaman kehadiran daring lebih cepat dibandingkan luring.

- (a) **Berapa detik perkiraan waktu interaksi yang Anda butuhkan untuk melakukan perekaman kehadiran daring di <https://akuhadir.unpar.ac.id?>**

Jumlah Responden	Waktu Perekaman Kehadiran Daring
1 orang	1 detik
1 orang	10 detik
2 orang	15 detik
1 orang	30 detik
1 orang	120 detik

Jika dihitung rata-rata waktu yang dibutuhkan untuk melakukan perekaman kehadiran daring bagi para dosen adalah 31,83 detik.

- (b) **Berapa detik perkiraan waktu interaksi yang Anda butuhkan untuk melakukan perekaman kehadiran luring menggunakan metode fingerprint?**

Jumlah Responden	Waktu Perekaman Kehadiran Luring
1 orang	1 detik
3 orang	5 detik
1 orang	40 detik
1 orang	90 detik

Jika dihitung rata-rata waktu yang dibutuhkan untuk melakukan perekaman kehadiran luring bagi para dosen adalah 24,33 detik.

Kesimpulan dari hasil survei dosen menunjukkan bahwa rata-rata waktu yang dibutuhkan secara luring adalah 24,33 detik lebih cepat dibandingkan dengan rata-rata waktu yang dibutuhkan secara daring

adalah waktu 31,83 detik.

##### 5. Menganalisis web *Student Portal UNPAR*.

**Status :** Ada sejak rencana kerja skripsi.

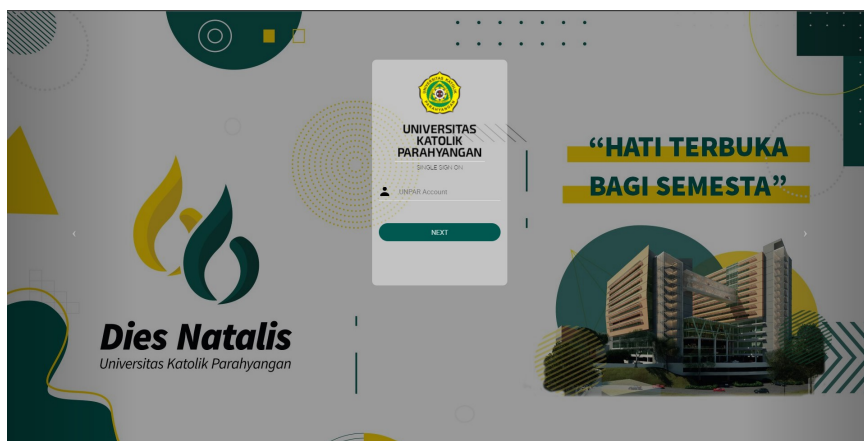
**Hasil :** Portal Akademik Mahasiswa adalah sebuah web yang di peruntukan bagi mahasiswa dalam rangka mendapatkan informasi kegiatan akademik mulai dari registrasi, melihat jadwal kuliah dan ujian, info nilai sampai pendaftaran sidang. Portal Akademik Mahasiswa dapat diakses melalui <https://studentportal.unpar.ac.id/>. Portal Akademik Mahasiswa Universitas Katolik Parahyangan yang terbaru sejak 2020 sudah dapat melakukan perekaman kehadiran secara online untuk setiap mata kuliah yang diambil. Analisis pada Portal Akademik Mahasiswa yang diutamakannya adalah bagaimana alur cara melakukan absensi pada mata kuliah. Berikut ini adalah alur untuk melakukan perekaman kehadiran online melalui Portal Akademik Mahasiswa Universitas Katolik Parahyangan:

- (a) Melakukan akses Portal Akademik Mahasiswa yang dapat diakses melalui <https://studentportal.unpar.ac.id/>.



Gambar 4: Tampilan halaman awal Portal Akademik Mahasiswa

- (b) Menekan tombol “Login” yang sudah tersedia agar dapat masuk ke dalam Portal Akademik Mahasiswa, dapat dilihat pada Gambar 4.
- (c) Memasukan *email* mahasiswa.

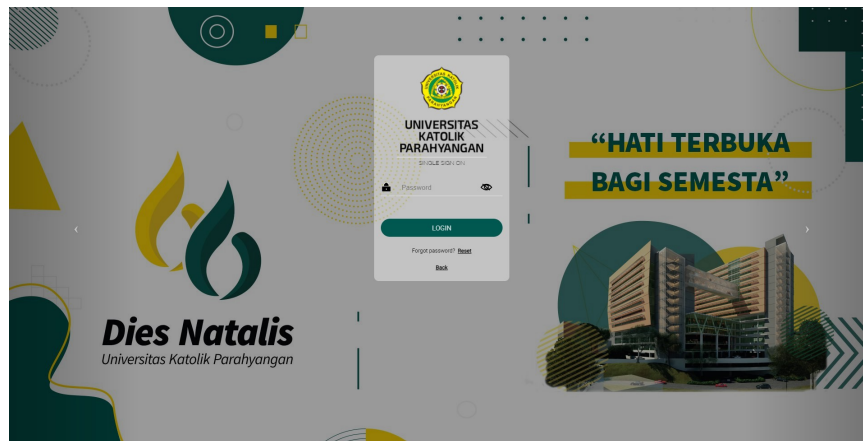


Gambar 5: Tampilan halaman Portal Akademik Mahasiswa untuk memasukan *email*

- (d) Menekan tombol “NEXT” setelah memasukan *email*, dapat dilihat pada Gambar 5.

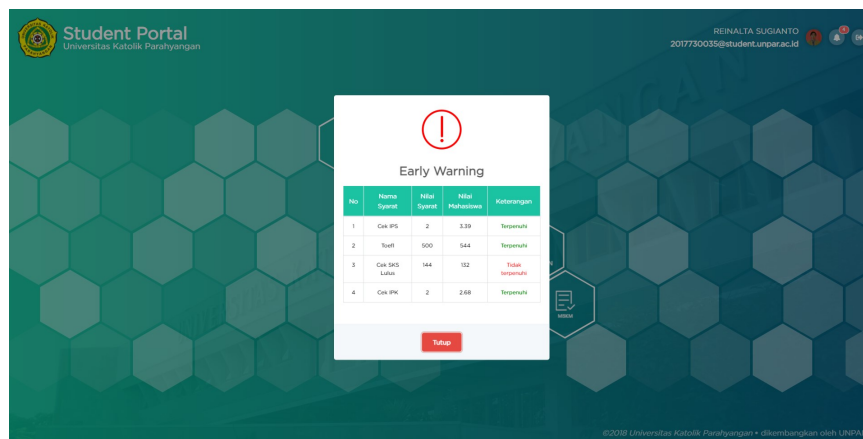


- (e) Memasukan *password* milik mahasiswa.

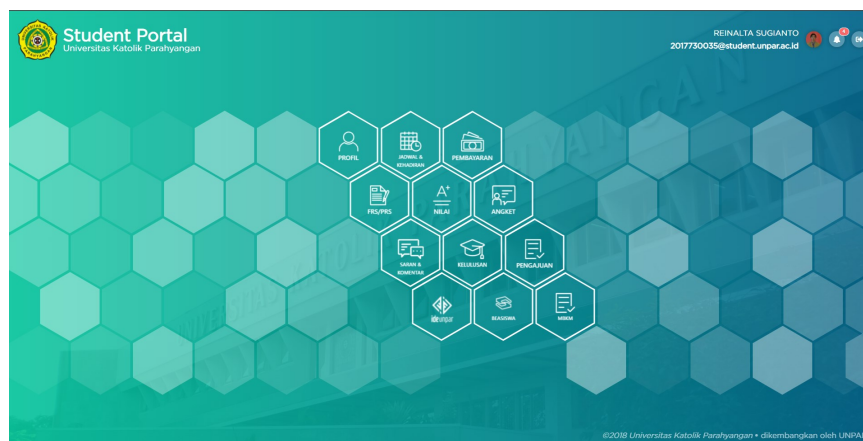


Gambar 6: Tampilan halaman Portal Akademik Mahasiswa untuk memasukan *password*

- (f) Menekan tombol “LOGIN” setelah memasukan *password*, dapat dilihat pada Gambar 6.
- (g) Menekan tombol “Tutup” jika muncul peringatan atau langsung menekan tombol berbentuk heksagon “JADWAL & KEHADIRAN” jika tidak muncul peringatan.



Gambar 7: Tampilan peringatan pada halaman Portal Akademik Mahasiswa

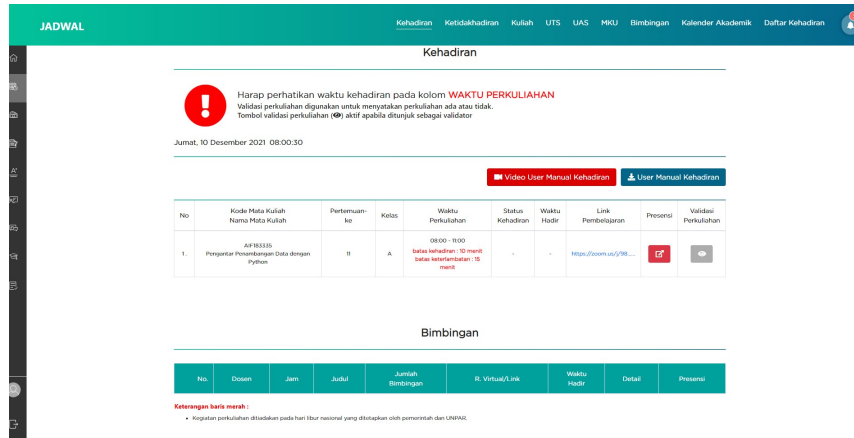


Gambar 8: Tampilan halaman Portal Akademik Mahasiswa setelah Berhasil *Login*

Pada Gambar 7 merupakan sebuah peringatan yang terkadang muncul menjelang berakhirnya

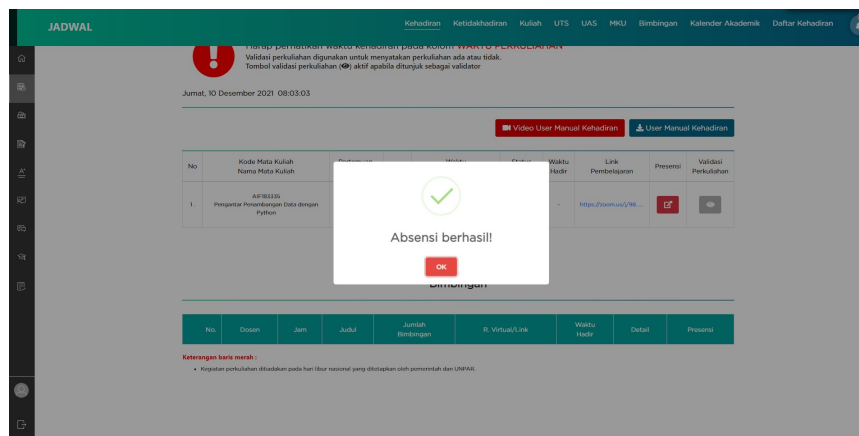
suatu semester untuk melihat status kebutuhan mahasiswa untuk lulus, sehingga perlu menekan tombol “Tutup” terlebih dahulu untuk menekan tombol berbentuk heksagon “JADWAL & KEHADIRAN” seperti pada Gambar 8. Jika tidak terjadi peringatan seperti pada Gambar 7, maka dapat langsung menekan tombol berbentuk heksagon “JADWAL & KEHADIRAN” seperti pada Gambar 8.

- (h) Menekan tombol berwarna merah pada kolom bagian persensi dari tabel jadwal kehadiran mata kuliah.



Gambar 9: Tampilan halaman Portal Akademik Mahasiswa untuk Melakukan Absen

- (i) Menekan tombol “OK” ketika muncul pemberitahuan setelah berhasil melakukan presensi.



Gambar 10: Tampilan Pemberitahuan Absensi Berhasil

## 6. Membangun program perekaman kehadiran daring otomatis.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Akan dilakukan pada Skripsi 2.

## 7. Melakukan pengujian dan eksperimen.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Akan dilakukan pada Skripsi 2.

## 8. Menulis dokumen skripsi.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Bab 1, Bab 2, Bab 3 sudah selesai ditulis, tetapi tidak menutup kemungkinan terjadi perubahan.

## 6 Pencapaian Rencana Kerja

Langkah-langkah kerja yang berhasil diselesaikan dalam Skripsi 1 ini adalah sebagai berikut:

1. Melakukan studi mengenai Selenium WebDriver.
2. Mempelajari bahasa pemrograman python.
3. Mempelajari cara menggunakan Selenium.
4. Melakukan survei ke beberapa mahasiswa Teknik Informatika UNPAR dan beberapa dosen Teknik Informatika UNPAR untuk mengetahui berapa lama waktu yang dibutuhkan untuk melakukan absensi secara daring dan luring.
5. Menganalisis web Student Portal UNPAR.
6. Menulis dokumen skripsi Bab 1-3.

Bandung, 13/12/2021

Reinalta Sugianto

Menyetujui,

Nama: Pascal Alfadian Nugroho  
Pembimbing Tunggal