

## BAB I

### PENDAHULUAN

Bab ini menjelaskan latar belakang dan rancangan solusi dari program Sistem Reservasi Fasilitas Umum Fakultas.

#### 1.1 Latar Belakang

Subbab ini menjelaskan latar belakang perancangan program Sistem Reservasi Fasilitas Umum Fakultas.

Fakultas memiliki berbagai fasilitas umum yang dapat digunakan oleh seluruh civitas akademika, seperti ruang kelas, laboratorium, ruang sidang, serta lapangan olahraga. Fasilitas-fasilitas ini penting untuk mendukung kegiatan belajar mengajar, penelitian, dan pengabdian kepada masyarakat.

Namun, seringkali terjadi *miss communication* dalam proses peminjaman fasilitas umum yang disebabkan oleh kurangnya sistem pemesanan yang jelas dan efisien. Hal ini dapat mengakibatkan terhambatnya kegiatan lain yang juga membutuhkan fasilitas tersebut. Selain itu, sistem pemesanan yang masih menggunakan berkas fisik rentan rusak dan hilang, sehingga dapat menimbulkan kesalahpahaman antara peminjam dan pihak yang bertanggung jawab.

Di era digital saat ini, diperlukan sistem pemesanan fasilitas umum yang lebih modern dan mudah digunakan. Sistem ini harus dapat mengintegrasikan seluruh fasilitas umum yang ada di fakultas, memungkinkan pengguna memesan secara langsung, dan menghasilkan *invoice* reservasi pemesanan.

#### 1.2 Rumusan Masalah

Subbab ini menjelaskan masalah yang mendasari program Sistem Reservasi Fasilitas Umum Fakultas.

- a. Setiap fakultas memiliki berbagai fasilitas umum yang bisa digunakan, tetapi perlu ada sistem yang jelas untuk memesan tempat/fasilitas umum sehingga menghindari tabrakan jadwal dengan kegiatan lain.



- b. Sistem yang disediakan harus jelas dan efisien yang membuat peminjam/pengguna fasilitas tidak kesulitan dalam melakukan peminjaman sehingga dapat meminimalisir juga munculnya kesalahpahaman antara peminjam fasilitas dan yang bertanggung jawab.
- c. Di era digital perlu ada sistem yang mempermudah pengguna dengan meminimalisir penggunaan berkas fisik yang rentan rusak dan mengalami kehilangan.

### 1.3 Tujuan

Subbab ini menjelaskan tujuan perancangan program Sistem Reservasi Fasilitas Umum Fakultas.

Berikut merupakan tujuan dalam perancangan program.

- a. Terbuatnya sistem pemesanan fasilitas yang terstruktur serta transparan sehingga setiap pemakaian fasilitas bisa terkoordinasi dengan baik dan menghindari benturan jadwal
- b. Sistem peminjaman fasilitas yang efisien berkembang, yang membuat pengguna dapat memesan fasilitas dengan mudah dan tanpa hambatan, serta mengurangi kemungkinan kesalahpahaman antara peminjam fasilitas dan pengelola fasilitas yang bertanggung jawab.
- c. Terciptanya sistem digital untuk pemesanan fasilitas yang mengurangi ketergantungan pada dokumen fisik, meningkatkan efisiensi administrasi, dan mengurangi risiko kerusakan dan kehilangan data lewat berkas digital.

### 1.4 Rancangan Solusi

Subbab ini menjelaskan rancangan solusi program Sistem Reservasi Fasilitas Umum Fakultas.

Program ini dirancang dengan rincian kebutuhan fungsional sebagai berikut:

- a. Membuat sistem yang terintegrasi pada suatu fakultas yang memiliki list atau pilihan fasilitas umum. Setelah melakukan pemesanan, tanggal yang sudah *dibooking* akan terdata, sehingga menghindari tabrakan jadwal.



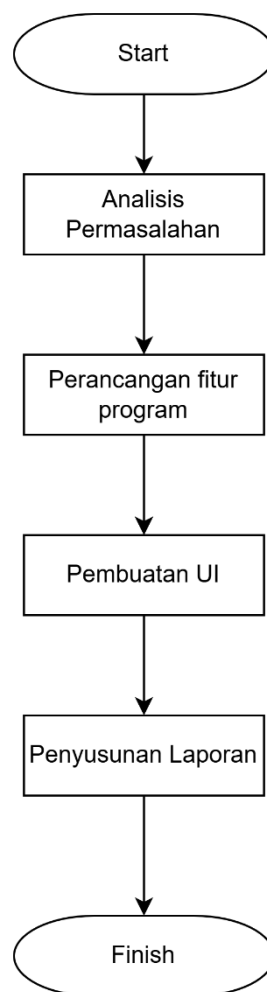
- b. Membuat sistem yang memungkinkan pengguna memesan keperluan pemesanan fasilitas umum secara langsung.
- c. Membuat sistem yang memberikan output invoice reservasi atas pemesanan fasilitas umum.

**BAB II****METODE PERANCANGAN**

Bab ini menjelaskan *flowchart* metode perancangan dan *flowchart* program Sistem Reservasi Fasilitas Umum Fakultas.

**2.1 Flowchart Metode Perancangan**

Subbab ini menjelaskan *flowchart* metode perancangan program Sistem Reservasi Fasilitas Umum Fakultas.



**Gambar 2.1** *Flowchart* Metode Perancangan Program Sistem Reservasi Fasilitas Umum Fakultas.



### 2.1.1. Analisis Permasalahan

Pada tahap ini dilakukan analisis permasalahan yang mendasari perancangan program Sistem Reservasi Fasilitas Umum Fakultas. Pada tahap ini kami mencari permasalahan yang terdapat pada kasus dalam pemilihan fasilitas karena sistem pemesanan yang ada saat ini masih memiliki beberapa kekurangan. Pertama, sistem pemesanan yang tidak jelas dan terpusat, sehingga sering terjadi tabrakan jadwal antara kegiatan yang berbeda. Kedua, sistem pemesanan yang rumit dan memakan waktu, sehingga pengguna merasa kesulitan dalam melakukan peminjaman. Ketiga, sistem pemesanan yang masih menggunakan berkas fisik, sehingga rentan rusak dan hilang.

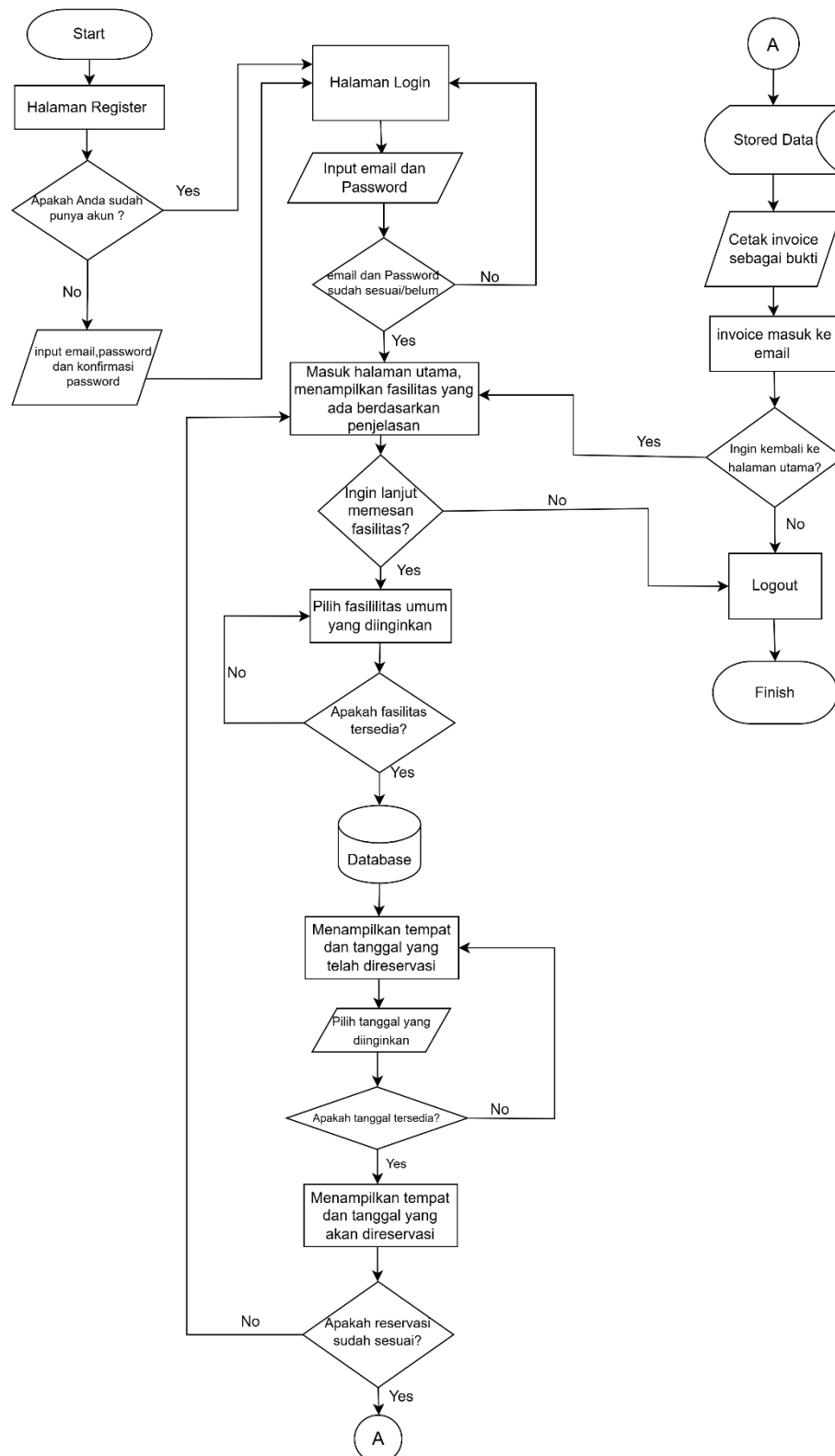
Permasalahan ini dapat diatasi dengan membangun sistem pemesanan yang terpusat, mudah diakses, dan efisien, serta berbasis digital. Sistem ini harus mampu mencegah tabrakan jadwal, memudahkan pengguna dalam melakukan peminjaman, dan memastikan akuntabilitas. Selain itu, perlu dilakukan edukasi dan sosialisasi kepada pengguna tentang cara menggunakan sistem pemesanan yang baru.

### 2.1.2. Analisis Flowchart Metode Perancangan

Pada saat pembuatan flowchart metode perancangan, langkah pertama kami menganalisis masalah yang sudah di rumuskan untuk di selesaikan yaitu dalam persoalan peminjaman fasilitas. Langkah berikutnya kami langsung menyusun langkah-langkah perancangan secara keseluruhan dimulai dari registrasi sampai pengiriman invoice melalui email.

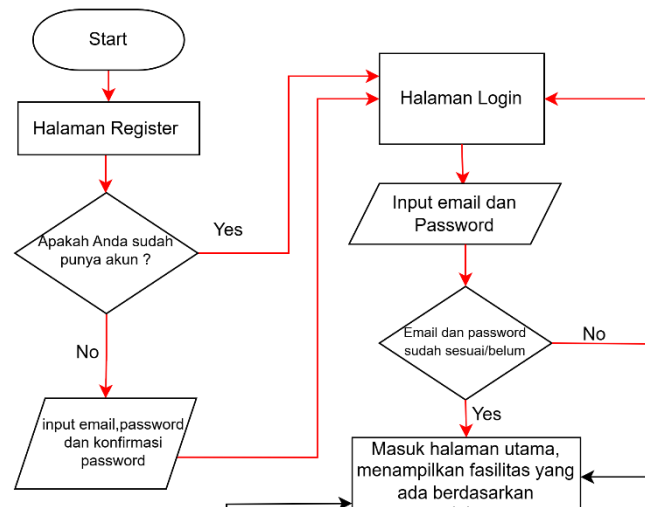
## 2.2 Flowchart Program

Subbab ini menjelaskan *flowchart* program Sistem Reservasi Fasilitas Umum Fakultas. Berikut merupakan *flowchart* program secara keseluruhan.



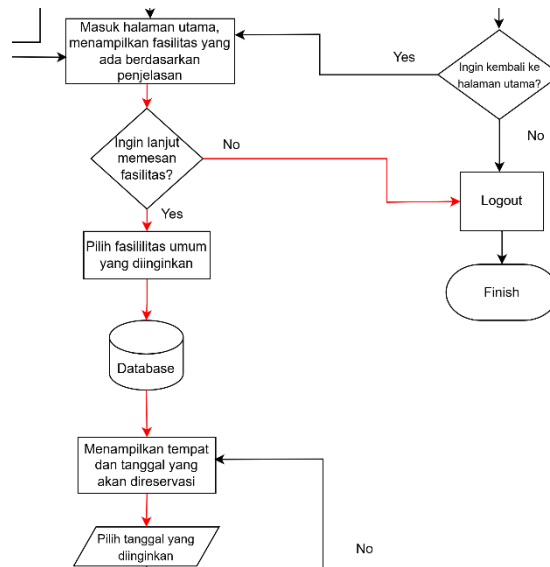
**Gambar 2.1.1** Flowchart Keseluruhan Program

*Flowchart* pada program kami diawali dengan memasuki halaman register yang dimana jika peminjam sudah memiliki akun akan diarahkan langsung untuk login dan menginput email serta password. Namun jika peminjam belum mempunyai akun maka akan diarahkan langsung untuk membuat akun dengan cara menginput email, password dan konfirmasi password sekali lagi setelah itu akan diarahkan kembali ke halaman login. Ketika login berhasil, peminjam akan langsung diarahkan ke halaman utama. Berikut merupakan *flowchart* fitur login



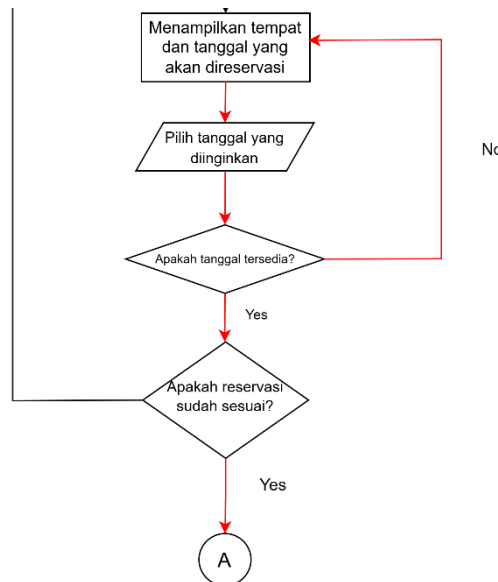
**Gambar 2.1.2** *Flowchart* Fitur Register & Login

Setelah masuk ke halaman utama, peminjam akan langsung disuguhkan pilihan ruangan yang ingin di pinjam. Jika peminjam ingin melanjutkan pemesanan maka program akan memunculkan pilihan fasilitas serta tanggal dan jam yang ingin di reservasi. Namun jika peminjam ingin membatalkan pemesanan maka program akan langsung menampilkan halaman logout.



**Gambar 2.1.3** Flowchart Fitur Pemesanan Fasilitas

Pada halaman peminjaman fasilitas, jika tanggal/jam yang dipilih sudah dipeservasi oleh peminjam lain maka program akan otomatis kembali ke halaman yang menampilkan tempat dan tanggal yang akan dipeservasi. Jika tanggal tersedia maka program akan memastikan kembali kepada peminjam apakah fasilitas yang di pinjam sudah sesuai.



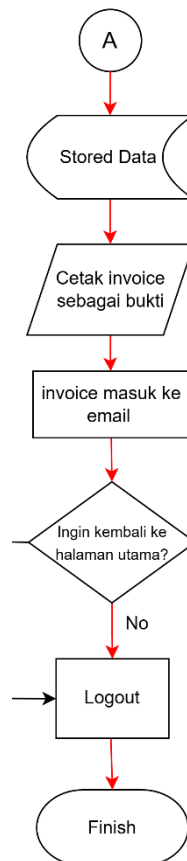
**Gambar 2.1.4** Flowchart Fitur Opsi Tanggal dan Jam

Jika peminjaman sudah sesuai maka program akan langsung mencetak invoice sebagai bukti. Program akan menampilkan opsi untuk kembali ke halaman



utama jika diinginkan, namun jika tidak maka program akan langsung menampilkan halaman logout.

Jika peminjaman sudah sesuai maka program akan langsung mencetak invoice sebagai bukti. Program akan menampilkan opsi untuk kembali ke halaman utama jika diinginkan, namun jika tidak maka program akan langsung menampilkan halaman logout.



**Gambar 2.1.5** *Flowchart Fitur Invoice*

## BAB III

### HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil dan pembahasan perancangan program Sistem Reservasi Fasilitas Umum Fakultas.

#### 3.1 Kode Program

Subbab ini menjelaskan fungsi yang digunakan pada program Sistem Reservasi Fasilitas Umum Fakultas

##### 3.1.1 Penggunaan Modul 'main\_login.py'

###### 1. Fungsi "button\_function"

Definisi ini sebagai fungsi untuk menjalankan perintah membaca database akun yang berupa file csv. Kemudian apabila program menemukan akun yang telah dimasukan pada database akun, maka program akan lanjut menuju program 'book\_fasilitas.py'. Sedangkan jika tidak ditemukan, maka program akan memberikan informasi bahwa pengguna belum memiliki akun. Selain itu, jika salah satu *input* yang dimasukan terdeteksi memiliki kesalahan, maka program akan memunculkan informasi bahwa *input* yang dimasukan memiliki kesalahan.

```
def button_function():
    # Check if the user exists in the database
    if os.path.exists("Akun.xlsx"):
        df = pd.read_excel("Akun.xlsx")
        email = entry1.get()
        password = entry2.get()

        # Validate the username and password
        if ((df['Email'] == email) & (df['Password'] == password)).any():
            with open("current_user.txt", "w") as file:
                file.write(email)
            app.destroy() # Destroy the current window
            import book_fasilitas
        else:
            messagebox.showerror("Error", "Email atau Password Tidak Valid")
    else:
        messagebox.showerror("Error", "You don't have an account!")
```

**Gambar 3.1.1.1** Fungsi "button\_function"

###### 2. Fungsi "belum\_punya\_akun"

Definisi ini digunakan sebagai fungsi untuk pembuatan akun bagi pengguna baru atau pengguna yang ingin mendaftar ( *sign up* ) dan

penyimpanan informasi pengguna seperti *email* dan *password* ke database melalui modul ‘register.py’

```
def belum_punya_akun():
    app.destroy() # Destroy the current window
    import register # Assuming register.py handles user registration
```

**Gambar 3.1.1.2** Fungsi “belum\_punya\_akun”

### 3. Fungsi “keluar”

Definisi ini berisi fungsi yang digunakan ketika dipanggil pada fungsi lainnya membuat program berhenti atau keluar.

```
def keluar():
    app.quit()
```

**Gambar 3.1.1.3** Fungsi “Keluar”

### 4. Fungsi untuk menampilkan GUI

Fungsi disini terdiri dari beberapa variabel yang memberikan *user interface* terhadap program ‘main\_login.py’. Fitur ‘frame’ pada program ini digunakan sebagai tempat bagi *widget* untuk menjalankan programnya. Fitur ‘entry’ pada program ini digunakan sebagai menerima *input* teks dari pengguna dan mengatur berbagai tampilan *widget* seperti panjang, lebar, *font*, warna latar belakang, dan warna teks. Fitur ‘button’ pada program ini digunakan untuk membuat tombol yang dapat diklik oleh pengguna untuk menjalankan perintah. Fitur ‘label’ pada program ini digunakan untuk menampilkan teks “Log In”.



```

1  img1 = ImageTk.PhotoImage(Image.open(
2      "Tugas Besar Reservasi Fasilitas Kelompok 5/Tugas Besar Reservasi Fasilitas/assets/bgbgt.png"))
3  l1 = customtkinter.CTkLabel(
4      master=app,
5      image=img1,
6      text="")
7  l1.pack()
8
9  # creating custom frame
10 frame = customtkinter.CTkFrame(
11     master=l1,
12     width=420,
13     height=460,
14     fg_color="#C8E2FF",
15     bg_color="#FDB484",
16     corner_radius=20)
17 frame.place(
18     relx=0.75,
19     rely=0.25,
20     anchor=tkinter.N)
21
22 l2 = customtkinter.CTkLabel(
23     master=frame,
24     text="Log In",
25     font=('Century Gothic', 44))
26 l2.place(x=75, y=35)
27
28
29 entry1 = customtkinter.CTkEntry(
30     master=frame,
31     width=270,
32     placeholder_text='Email',
33     font=("Century Gothic", 22))
34 entry1.place(x=75, y=110)
35
36 entry2 = customtkinter.CTkEntry(
37     master=frame,
38     width=270,
39     placeholder_text='Password',
40     font= ("Century Gothic", 22),
41     show="*")
42 entry2.place(x=75, y=165)
43
44 l3 = customtkinter.CTkButton(
45     master=frame,
46     text="Belum Punya Akun?",
47     font=('Century Gothic', 14),
48     fg_color='#C8E2FF',
49     hover_color='#C8E2FF',
50     text_color="Black",
51     command=belum_punya_akun)
52 l3.place(x=180, y=340)
53
54 # Create custom button
55 button1 = customtkinter.CTkButton(
56     master=frame,
57     width=270,
58     text="Login",
59     font=("Century Gothic", 24),
60     command=button_function,
61     corner_radius=6)
62 button1.place(x=75, y=300)
63
64
65
66 imgexit = Image.open("Tugas Besar Reservasi Fasilitas Kelompok 5/Tugas Besar Reservasi Fasilitas/assets/logo.png")
67
68 btnexit = CTkButton(
69     master=app,
70     text="Keluar",
71     corner_radius=32,
72     height=30,
73     fg_color="#F80000",
74     hover_color="#8D0000",
75     border_color="#C8E2FF",
76     border_width=2,
77     image=CTkImage(dark_image=imgexit, light_image=imgexit),
78     bg_color="#FDB484",
79     command=keluar,
80     font=("Arial", 18))
81 btnexit.place(relx=0.98, rely=0.07, anchor="se")
82
83 app.mainloop()

```

**Gambar 3.1.1.4** Fungsi “untuk menampilkan GUI”

### 3.1.2 Penggunaan Modul 'register.py'

#### 1. Fungsi "register\_user"

Definisi ini berisi fungsi yang digunakan pada proses di mana pengguna baru membuat akun pada program. Proses ini melibatkan pengumpulan data pengguna dan penyimpanan informasi pengguna ke dalam database. Proses diawali dengan pengguna memasukkan *email*, *password* dari pengguna dan dibaca program, serta dilanjut untuk pencatatan ke dalam database yang berupa file csv.

```
def register_user():
    email = entry1.get()
    password = entry2.get()
    confirm_password = entry3.get()

    if password == confirm_password:
        if os.path.exists("Akun.xlsx"):
            df = pd.read_excel("Akun.xlsx")
        else:
            df = pd.DataFrame(columns=['Email', 'Password'])

        if email in df['Email'].values:
            messagebox.showerror("Error", "User sudah ada!")
        else:
            new_user = pd.DataFrame({'Email': [email], 'Password': [password]})
            df = pd.concat([df, new_user], ignore_index=True)
            df.to_excel("Akun.xlsx", index=False)
            messagebox.showinfo("Success", "Registrasi Berhasil!")
            app.destroy()
            import main_login
    else:
        messagebox.showerror("Error", "Password tidak sama!")
```

Gambar 3.1.1.5 Fungsi "register\_user"

#### 2. Fungsi "keluar"

Definisi ini berisi fungsi yang digunakan ketika dipanggil pada fungsi lainnya membuat program berhenti atau keluar.

```
1 def keluar():
2     app.destroy()
3     import main_login as ml
4     ml.login_main()
5     return
```

Gambar 3.1.1.6 Fungsi "keluar"

### 3.1.3 Penggunaan Modul 'book\_fasilitas.py'

#### 1. Fungsi "show\_popup"

Definisi ini digunakan untuk menampilkan pesan kepada pengguna dalam sebuah jendela kecil yang muncul di atas jendela utama. Pada program ini menampilkan gambar salah satu fasilitas yang



dipilih pada fungsi sebelumnya beserta dengan deskripsi dari fasilitas tersebut. Didalamnya membuat window baru dengan fitur customtkinter 'TopLevel' guna menampilkannya diatas program utama 'book\_fasilitas'. Kemudian *widget-widget* lain pada *library* customtkinter seperti Label, Frame, dan Button digunakan untuk mengintegrasikannya dengan fungsi-fungsi yang lainnya.



```

1 def show_popup(image_path, description, module_name, facility_name):
2     popup = CTkTopLevel(app)
3     popup.attributes("-fullscreen", True)
4     popup.title("Detail Fasilitas")
5     popup.resizable(True, True)
6
7     imgpopup = ImageTk.PhotoImage(Image.open("Tugas Besar Reservasi Fasilitas Kelompok 5/Tugas Besar Reservasi Fasilitas/assets/bglain.png"))
8     l1 = customtkinter.CTkLabel(master=popup, image=imgpopup, text="")
9     l1.pack()
10
11     framepopup = customtkinter.CTkFrame(
12         master=l1,
13         width=500,
14         height=510,
15         fg_color="#9FC0FF",
16         bg_color="#F0B484")
17     framepopup.place(relx=0.50, rely=0.50, anchor=tk.CENTER)
18
19     # Ensure the popup stays in front
20     popup.transient(app)
21     popup.grab_set()
22
23     img = Image.open(image_path)
24     img = img.resize((500, 500), Image.LANCZOS)
25     img_tk = ImageTk.PhotoImage(img)
26
27     frame1 = customtkinter.CTkFrame(
28         master=popup,
29         width=270,
30         height=270,
31         fg_color="#9FC0FF")
32     frame1.place(
33         relx=0.5,
34         rely=0.6,
35         anchor=tk.S)
36
37     # Keep a reference to prevent garbage collection
38     label_image = CTkLabel(master=popup, image=img_tk, text="")
39     label_image.image = img_tk # Keep a reference to prevent garbage collection
40     label_image.pack(pady=20)
41     label_image.place(relx=0.5, rely=0.61, anchor=tk.S)
42
43     frame2 = customtkinter.CTkFrame(
44         master=popup,
45         width=500,
46         height=65,
47         fg_color="#9FC0FF",
48         corner_radius=0)
49     frame2.place(
50         relx=0.5,
51         rely=0.615,
52         anchor=tk.N)
53
54     header_label = CTkLabel(
55         master=popup,
56         text=facility_name,
57         font=("Rockwell", 42, "bold"),
58         text_color="black",
59         fg_color="#F0B484")
60     header_label.pack(pady=10)
61     header_label.place(relx=0.5, rely=0.075, anchor="n")
62
63     label_description = CTkLabel(
64         master=popup,
65         text=description,
66         wraplength=500,
67         font=("Comic Sans MS", 22),
68         text_color="black",
69         fg_color="#9FC0FF")
70     label_description.pack(pady=20)
71     label_description.place(relx=0.5, rely=0.62, anchor="n")
72
73     # Keep a reference to prevent garbage collection
74     popup.image = img_tk
75
76     img3 = Image.open("Tugas Besar Reservasi Fasilitas Kelompok 5/Tugas Besar Reservasi Fasilitas/assets/logo.png")
77
78     def book():
79         try:
80             app.destroy()
81             module = importlib.import_module(module_name)
82             if hasattr(module, 'start_app'):
83                 module.start_app()
84             else:
85                 print(f"Module {module_name} does not have a start_app function.")
86         except ImportError:
87             print(f"Module {module_name} tidak ditemukan.")
88
89     btn = CTkButton(
90         master=popup,
91         text="Booking Sekarang",
92         corner_radius=32,
93         fg_color="#222083",
94         hover_color="#C850C0",
95         border_color="#FFC070",
96         border_width=2,
97         bg_color="#F0B484",
98         image=CTkImage(dark_image=img3, light_image=img3),
99         font=("Comic Sans MS", 24),
100         command=book)
101     btn.place(relx=0.5, rely=0.85, anchor="n")
102
103     # Define the exit button
104     exitbtn = customtkinter.CTkButton(
105         master=popup,
106         text="Kembali",
107         corner_radius=32,
108         fg_color="#F0B484",
109         hover_color="#808080",
110         border_color="#C850C0",
111         border_width=2,
112         bg_color="#F0B484",
113         image=CTkImage(dark_image=img3, light_image=img3),
114         font=("Arial", 18),
115         command=popup.destroy)
116     exitbtn.place(
117         relx=0.08,
118         rely=0.07,
119         anchor="se")

```

**Gambar 3.1.2.1** Fungsi “Show pop-up”

## 2. Fungsi “book”

Definisi ini merupakan fungsi lanjutan dari fungsi show pop-up yang digunakan pengguna untuk melakukan pemesanan fasilitas pada program ini. Selain itu, fungsi ini juga digunakan untuk memuat ulang modul tanpa harus memulai ulang keseluruhan program.

```
def book():
    try:
        app.destroy()
        module = importlib.import_module(module_name)
        if hasattr(module, 'start_app'):
            module.start_app()
        else:
            print(f"Module {module_name} does not have a start_app function.")
    except ImportError:
        print(f"Module {module_name} tidak ditemukan.")
```

**Gambar 3.1.2.2** Fungsi “Book”

### 3.1.4 Penggunaan Modul Fasilitas

(‘lapangan.py’;’ruang\_multimedia.py’;’ruang\_seminar.py’;’working\_space.py’)

#### 1. Fungsi “get\_user\_email”

Fungsi ini digunakan untuk membaca email pengguna dari sebuah file yang bernama ‘current\_user.txt’. Jika file tersebut tidak ditemukan, fungsi akan menampilkan pesan kesalahan dan menutup aplikasi.

```
def get_user_email(self):
    try:
        with open('current_user.txt', 'r') as file:
            email = file.readline().strip()
            return email
    except FileNotFoundError:
        messagebox.showerror("Error", "current_user.txt not found!")
        self.root.destroy()
```

**Gambar 3.1.2.3** Fungsi “Get user email”

#### 2. Fungsi “see\_date”

Fungsi ini digunakan untuk mengelola dan memproses pemesanan fasilitas. Fungsi ini memeriksa ketersediaan slot pemesanan dan menampilkan statusnya kepada pengguna, serta menyimpan informasi pemesanan. Apabila slot tersedia maka setelah memilih tanggal, waktu, dan durasi maka akan memunculkan tulisan ‘Slot already booked (on date) at (start time)’ sesuai dengan tanggal dan waktu yang dipilih.



```
def see_date(self):
    date = self.cal.get()
    start_time = self.time_combobox.get()
    duration = int(self.duration_combobox.get())
    if self.is_booked(date, start_time, duration):
        self.date_label.configure(text=f"Slot already booked on {date} at {start_time}", text_color='red')
    else:
        booking_info = f"{date} at {start_time} for {duration} hours"
        self.date_label.configure(text=booking_info, text_color='#2a9fd6')
        self.save_to_csv(date, start_time, duration)
        self.generate_invoice(date, start_time, duration)
        self.update_available_times()
```

**Gambar 3.1.2.4** Fungsi “See date”

### 3. Fungsi “save\_to\_csv”

Fungsi ini digunakan untuk menyimpan informasi pemesanan data nama, email, tanggal, waktu, dan durasi yang telah dipilih oleh pengguna ke database pada file CSV yang bernama ‘booking\_data’. Jika file CSV belum ada, fungsi ini juga akan membuat file baru.

```
def save_to_csv(self, date, start_time, duration):
    file_path = f'{self.facility_name}_booking_data.csv'
    file_exists = os.path.isfile(file_path)

    name = self.entry_name.get()
    email = self.user_email

    with open(file_path, mode='a', newline='') as file:
        writer = csv.writer(file)
        if not file_exists:
            writer.writerow(["Date", "Start Time", "Duration (hours)", "Name", "Email"])
        writer.writerow([date, start_time, duration, name, email])
```

**Gambar 3.1.2.5** Fungsi “Save to CSV”

### 4. Fungsi “update\_available\_times”

Fungsi ini digunakan untuk memperbarui daftar waktu yang tersedia berdasarkan tanggal yang dipilih dan data pemesanan yang sudah ada. Fungsi ini membaca data dari file CSV, mengidentifikasi waktu yang sudah dipesan untuk tanggal yang dipilih, dan memperbarui combobox waktu agar menampilkan ketersediaan yang sebenarnya.

```
def update_available_times(event=None):
    date = cal.get()
    unavailable_times = set()

    file_path = f'{facility_name}_booking_data.csv'
    if os.path.exists(file_path):
        with open(file_path, mode='r', newline='') as file:
            reader = csv.reader(file)
            next(reader)
            for row in reader:
                booked_date, booked_time, booked_duration = row[:3]
                if booked_date == date:
                    booked_start_hour = int(booked_time.split(":")[0])
                    booked_end_hour = booked_start_hour + int(booked_duration)
                    unavailable_times.update(range(booked_start_hour, booked_end_hour))

    updated_time_values = []
    for time in time_values:
        hour = int(time.split(":")[0])
        if hour in unavailable_times:
            updated_time_values.append(f"{time} (booked)")
        else:
            updated_time_values.append(time)

    time_combobox.configure(values=updated_time_values)
    time_combobox.set(updated_time_values[0])
```

**Gambar 3.1.2.6** Fungsi “update\_available\_times”

## 5. Fungsi “is\_booked”

Fungsi ini digunakan untuk memeriksa apakah slot waktu tertentu pada tanggal tertentu sudah dipesan dengan memeriksa data dalam file CSV. Fungsi ini penting untuk memastikan bahwa pengguna tidak dapat memesan slot yang sudah ditempati oleh pengguna lain.

```
def is_booked(date, start_time, duration):
    file_path = f'{facility_name}_booking_data.csv'
    if not os.path.exists(file_path):
        return False

    start_hour = int(start_time.split(":")[0])
    booked_hours = set(range(start_hour, start_hour + duration))

    with open(file_path, mode='r', newline='') as file:
        reader = csv.reader(file)
        next(reader)
        for row in reader:
            booked_date, booked_time, booked_duration = row[:3]
            if booked_date == date:
                booked_start_hour = int(booked_time.split(":")[0])
                booked_end_hour = booked_start_hour + int(booked_duration)
                if booked_hours & set(range(booked_start_hour, booked_end_hour)):
                    return True
    return False
```

**Gambar 3.1.2.7** Fungsi “is booked”

## 6. Fungsi “justify\_text”

Fungsi ini digunakan untuk menulis teks ke dalam PDF dengan format kanan-kiri (justify), memastikan bahwa setiap baris teks memiliki lebar yang sama. Fungsi ini mempertimbangkan pemisahan

baris berdasarkan lebar tertentu yang diinginkan dan membantu dalam menjaga tata letak teks yang rapi dalam dokumen PDF

```
def justify_text(pdf, text, x, y, width):
    lines = text.split("\n")
    for line in lines:
        words = line.split()
        if len(words) == 0:
            pdf.ln(10)
            continue

        space_width = pdf.get_string_width(' ')
        line_width = sum(pdf.get_string_width(word) for word in words) + space_width * (len(words) - 1)

        if line_width <= width:
            pdf.set_xy(x, y)
            pdf.multi_cell(width, 10, line, align='J')
            y += 10
            continue

        output = []
        while words:
            while words and sum(pdf.get_string_width(word) for word in output + [words[0]]) + space_width * len(output) <= width:
                output.append(words.pop(0))

            if words:
                total_spaces = len(output) - 1
                if total_spaces > 0:
                    extra_space = (width - sum(pdf.get_string_width(word) for word in output)) / total_spaces
                else:
                    extra_space = 0

                for i in range(len(output) - 1):
                    output[i] += ' ' * (int(extra_space / space_width))

            pdf.set_xy(x, y)
            pdf.multi_cell(width, 10, ' '.join(output), align='J')
            y += 10
            output = []

        if output:
            pdf.set_xy(x, y)
            pdf.multi_cell(width, 10, ' '.join(output), align='J')
            y += 10
```

**Gambar 3.1.2.8** Fungsi “justify text”

## 7. Fungsi “save\_invoice”

Fungsi ini digunakan untuk membuat dan menyimpan invoice dengan format terstruktur dalam format PDF. Fungsi ini menggunakan library ‘FPDF’ untuk pembuatan PDF dan ‘messagebox’ untuk menampilkan notifikasi kepada pengguna. Hasilnya disimpan sebagai file PDF dengan nama yang disesuaikan berdasarkan nama pengguna, tanggal, waktu, dan durasi.



```
def save_invoice(text, name, date):
    filename = f"Invoice_{name}_{date.replace('/', '-')}.pdf"
    pdf = FPDF()
    pdf.add_page()

    pdf.set_font("Arial", 'B', 30)
    pdf.cell(0, 30, "UnitySpace", ln=True, align='C')

    pdf.set_font("Arial", size=12)

    text_width = 100
    text_height = 70
    page_width = 210
    border_x = (page_width - text_width) / 2
    border_y = 45

    pdf.set_draw_color(0, 0, 0)
    pdf.set_fill_color(255, 255, 255)
    pdf.rect(border_x, border_y, text_width, text_height, 'D')

    text_x = border_x + 2
    text_y = border_y + 2
    justify_text(pdf, text, text_x, text_y, text_width - 4)

    if os.path.exists('Tugas Besar Reservasi Fasilitas/assets/logo.png'):
        logo_width = 35
        logo_x = (page_width - logo_width) / 2
        pdf.image('Tugas Besar Reservasi Fasilitas/assets/logo.png', x=logo_x, y=border_y + text_height + 10, w=logo_width)

    pdf.output(filename)
    messagebox.showinfo("Invoice Generated", f"Invoice saved as {filename}")
    return filename
```

**Gambar 3.1.2.9** Fungsi “save invoice”

## 8. Fungsi “send\_email\_with\_invoice”

Fungsi ini digunakan untuk mengirim email dengan lampiran invoice berupa PDF kepada pemesan. Fungsi ini menggunakan SMTP untuk mengirim email melalui server Gmail dengan memastikan bahwa file PDF dilampirkan dan dikirim dengan benar.

```
def send_email_with_invoice(recipient_email, invoice_filename):
    sender_email = "kelompok5prakprokom@gmail.com"
    sender_password = "tvsh fnti qfgt eieq"

    subject = "Invoice Reservasi Lapangan"
    body = "Berikut ini kami kirimkan invoice reservasi fasilitas Lapangan yang telah anda pesan."

    msg = MIMEMultipart()
    msg['From'] = sender_email
    msg['To'] = recipient_email
    msg['Subject'] = subject
    msg.attach(MIMEText(body, 'plain'))

    with open(invoice_filename, "rb") as attachment:
        part = MIMEBase('application', 'octet-stream')
        part.set_payload(attachment.read())
        encoders.encode_base64(part)
        part.add_header(
            'Content-Disposition',
            f'attachment; filename={invoice_filename}',
        )
    msg.attach(part)

    try:
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(sender_email, sender_password)
        text = msg.as_string()
        server.sendmail(sender_email, recipient_email, text)
        server.quit()
        messagebox.showinfo("Email Sent", f"Invoice emailed to {recipient_email}")
    except Exception as e:
        messagebox.showerror("Error", f"Failed to send email: {str(e)}")
```

**Gambar 3.1.2.10** Fungsi “send email with invoice”

## 9. Fungsi “balikmilih”

Fungsi ini digunakan untuk menutup program yang sedang berjalan dan kembali ke modul sebelumnya yaitu ‘book\_fasilitas’ untuk memilih fasilitas.

```
def balikmilih():
    app.destroy()
    import book_fasilitas
    return
```

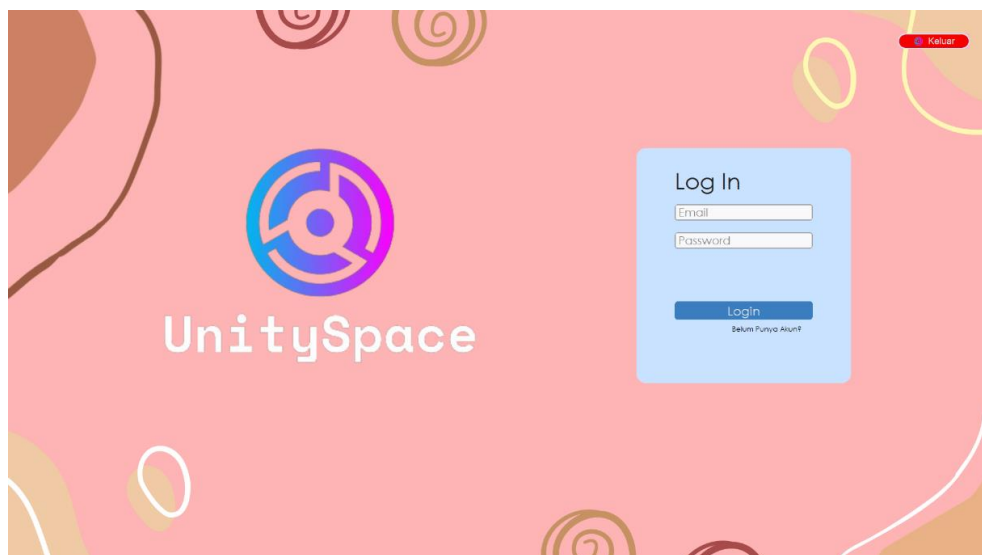
**Gambar 3.1.2.11** Fungsi “balikmilih”

### 3.2 User Interface (UI)

Subbab ini menjelaskan tampilan program Sistem Reservasi Fasilitas Umum Fakultas pada saat program dijalankan.

1. Saat program dijalankan, jendela GUI program login ditampilkan. Pengguna dapat memasukkan email dan password untuk login, atau memilih untuk mendaftar akun baru jika belum memiliki akun dengan menekan tombol "Belum punya akun?" maka akan ditampilkan halaman register. Jika login berhasil, aplikasi akan mengarah ke modul 'book\_fasilitas'. Selain itu, pada halaman login ini terdapat tombol di pojok kanan atas untuk keluar dari aplikasi. Apabila tombol "Keluar" ditekan aplikasi akan tertutup.

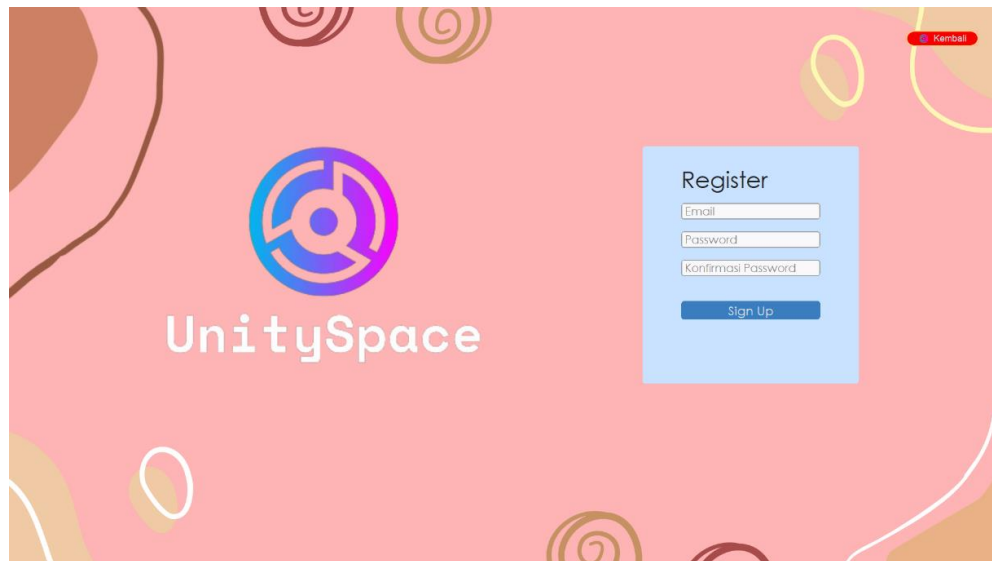
Berikut tampilan UI setelah program dijalankan:



**Gambar 3.2.1** Tampilan "Menu Login"

2. Saat program dijalankan, jendela registrasi aplikasi GUI ditampilkan. Pengguna dapat memasukkan email, password, dan konfirmasi password, lalu menekan tombol "Sign Up" untuk mendaftarkan diri sebagai pengguna. Jika registrasi berhasil akan memunculkan pesan "Registrasi telah berhasil" dan data pengguna akan disimpan ke file 'Akun.xlsx' dan aplikasi akan beralih ke halaman login. Jika pengguna menekan tombol "Kembali", aplikasi akan kembali ke halaman login tanpa menyimpan data.

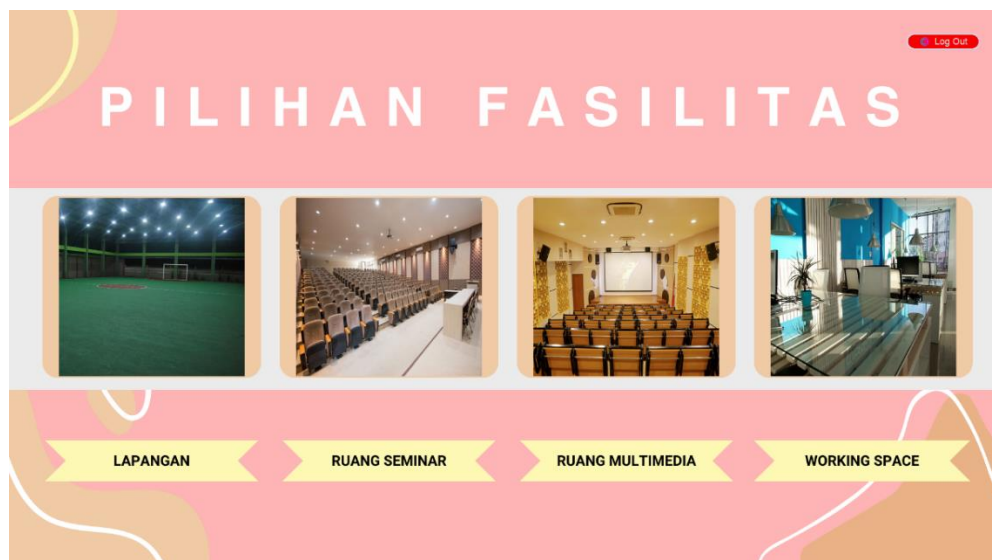
Berikut tampilan UI setelah program dijalankan:



**Gambar 3.2.2** Tampilan “Menu Register”

3. Saat program dijalankan akan menampilkan pilihan empat fasilitas yaitu lapangan, ruang seminar, ruang multimedia, dan working space beserta gambarnya masing-masing. Ketika pengguna menekan tombol fasilitas, sebuah jendela pop-up muncul dengan detail lebih lanjut tentang fasilitas tersebut dan opsi untuk melakukan pemesanan dengan tombol ”Booking sekarang”. Program ini juga menyediakan tombol keluar. Apabila tombol keluar ditekan maka akan kembali ke halaman login.

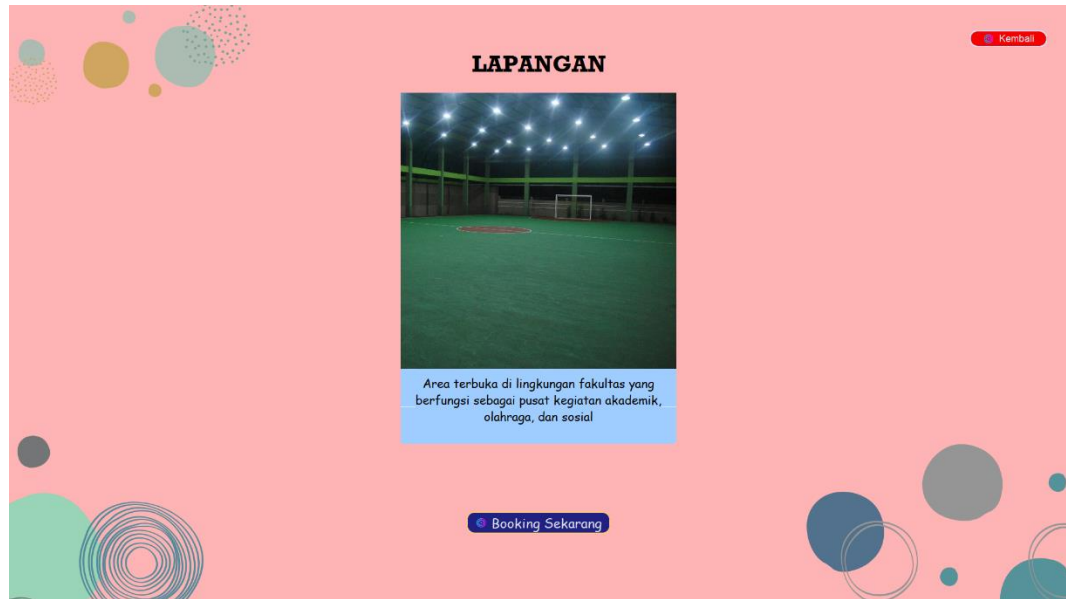
Berikut tampilan UI setelah program dijalankan:



**Gambar 3.2.3** Tampilan “Menu Pilihan Fasilitas”

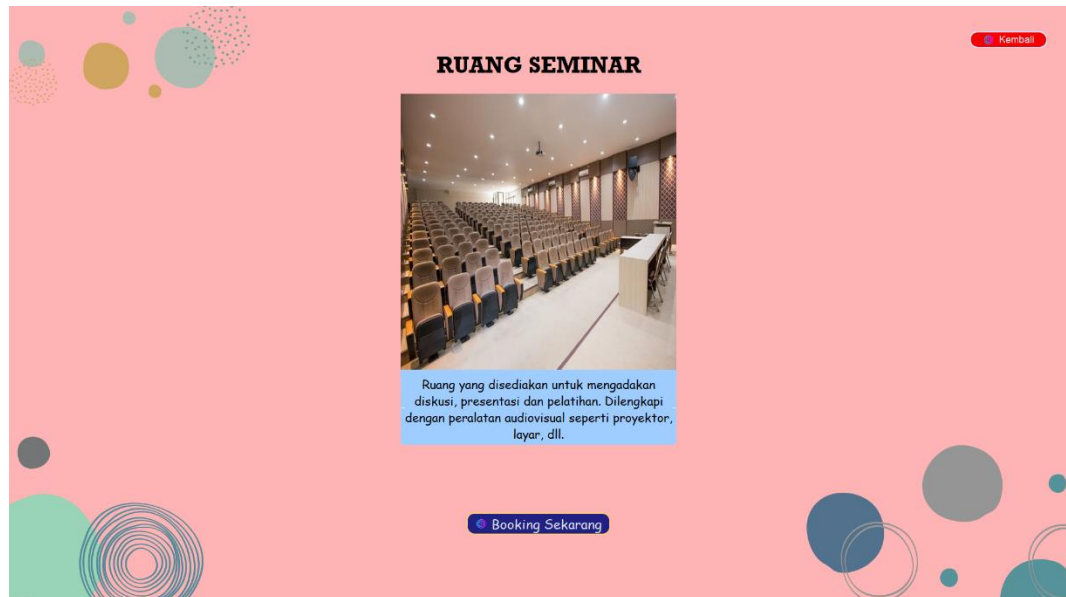


Berikut tampilan UI setelah menekan gambar fasilitas lapangan:



**Gambar 3.2.4** Tampilan “Fasilitas Lapangan”

Berikut tampilan UI setelah menekan gambar fasilitas ruang seminar:



**Gambar 3.2.5** Tampilan “Fasilitas Ruang Seminar”

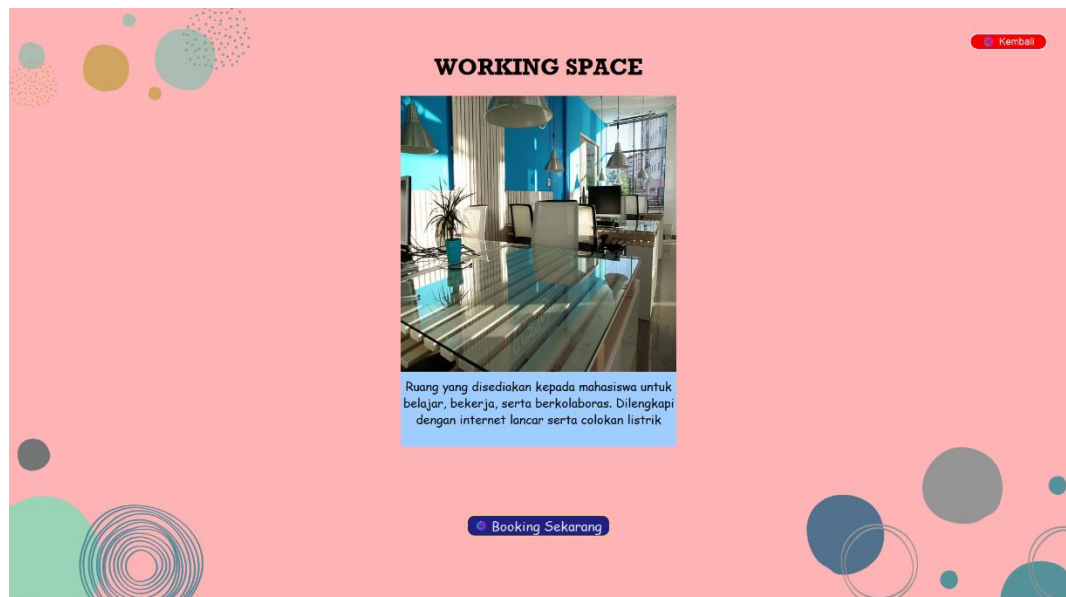


Berikut tampilan UI setelah menekan gambar fasilitas ruang multimedia:



**Gambar 3.2.6** Tampilan “Fasilitas Ruang Multimedia”

Berikut tampilan UI setelah menekan gambar fasilitas working space:



**Gambar 3.2.7** Tampilan “Fasilitas Working Space”

4. Saat program dijalankan akan menampilkan kalender, pilihan jam pemesanan, dan pilihan durasi waktu pemesanan. Email yang ditampilkan merupakan email yang digunakan pada saat login dan tidak bisa diganti. Pengguna dapat mengisi data nama, kemudian dapat memilih tanggal pemesanan sesuai yang diinginkan. Selanjutnya, pengguna dapat memilih jam pemesanan. Apabila tanggal dan jam yang dipilih tersedia maka data akan disimpan ke dalam CSV. Namun, apabila tanggal dan jam yang dipilih tidak

tersedia maka akan menampilkan pesan bahwa slot sudah terisi. Selanjutnya, pengguna dapat memilih durasi waktu yang diinginkan. Jika data sudah terisi dengan benar, pengguna dapat menekan tombol 'Save Booking'. Data yang telah dimasukkan oleh pengguna akan tersimpan ke database sesuai dengan pilihan fasilitas tersebut dalam bentuk file CSV. Setelah data tersimpan, akan memunculkan pesan "data telah tersimpan" dan *invoice* di kirim ke email pengguna yang digunakan untuk login.

**Gambar 3.2.8** Tampilan “Menu Reservasi”

5. Berikut merupakan tampilan *invoice* setelah pengguna berhasil memesan fasilitas pada program. *Invoice* disimpan secara lokal juga dikirimkan melalui *email* yang sama dengan akun pengguna yang digunakan untuk memesan.

## UnitySpace

Nama : faiq  
Email : dimasfaiqwiraanwar@gmail.com  
Fasilitas : Ruang Seminar  
Tanggal : 31/05/24  
Waktu : 09:00  
Durasi : 1 jam



**Gambar 3.2.9** Tampilan *Invoice* Program Pemesanan

**BAB IV****PENUTUP**

Kesimpulan dari hasil perancangan program Sistem Reservasi Fasilitas Umum Fakultas adalah sebagai berikut:

1. Sistem reservasi fasilitas terstruktur dan transparan yang telah dibuat memungkinkan semua penggunaan sistem dikoordinasikan dengan tepat dan menghindari tabrakan antar jadwal. Sistem mencatat semua pesanan, sehingga jadwal pemesanan fasilitas dapat terlihat dan terhindar dari tabrakan.
2. Sistem peminjaman fasilitas efisien dan mudah digunakan yang diciptakan dapat mengurangi hambatan bagi pengguna saat memesan fasilitas. Sistem ini juga meminimalkan kesalahpahaman antara peminjam dan pengelola fasilitas, sehingga membuat proses peminjaman lebih lancar dan transparan. Pengguna dapat melakukan reservasi fasilitas secara langsung dan mudah tanpa melalui prosedur yang rumit.
3. Penerapan sistem reservasi fasilitas digital yang telah dibuat dapat mengurangi ketergantungan pada dokumen fisik, meningkatkan efisiensi administrasi, dan mengurangi risiko kerusakan atau kehilangan data. Sistem memberikan output dalam bentuk *invoice* reservasi digital, sehingga memudahkan pelacakan dan pengelolaan reservasi fasilitas.