

BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Seperti yang telah dijelaskan pada bab sebelumnya, pelaksanaan magang ini ditempatkan pada divisi Engineering mengisi posisi *Front-end Developer Intern*. Pelaksanaan magang sebagai *Front-end Developer Intern* di perusahaan ini dibimbing oleh Robin Nusantara yang menjabat sebagai *Back-end Developer* di perusahaan ini. Selama pelaksanaan magang di perusahaan ini, semua tugas yang didapatkan dapat berasal dari siapa saja, baik dari *user*, divisi lain, ataupun anggota Engineering, selama tugas yang diberikan dianggap sesuai dengan keahlian. Proses pengerjaan tugas nantinya dilaporkan setiap harinya ketika *Alignment Sync Up*. Tugas "Promenade" diberikan oleh Felix Ivander yang merupakan *Data Scientist* yang menjadi salah satu proyek yang dikerjakan selama keberlangsungan magang di perusahaan ini.

3.2 Tugas yang Dilakukan

Tugas dan tanggung jawab utama selama pelaksanaan kerja magang di Bukit Vista Hospitality Services sebagai *Frontend Developer Intern* adalah mengerjakan proyek Promenade yang dibangun selama pelaksanaan magang ini dilakukan.

Berikut merupakan uraian tugas yang dilakukan selama pelaksanaan magang.

1. Mengembangkan *Front-End* pada website BIGRR
2. Mengembangkan dan membangun beberapa fitur pada Slack Bot GAIA
3. Membangun website dasbor Promenade
4. Melakukan testing pada setiap fitur atau perbaikan yang telah dilakukan
5. Melakukan pemecahan masalah, debug, dan *bugfixing* perangkat lunak.

3.3 Uraian Pelaksanaan Magang

Berikut adalah uraian pekerjaan yang dilakukan selama melaksanakan kerja magang di perusahaan Bukit Vista Hospitality Services sebagai *Frontend Developer*

Intern yang dilakukan selama 3 bulan. Dapat dilihat pada tabel 3.1

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama magang

Minggu Ke-	Pekerjaan yang dilakukan
1	<ul style="list-style-type: none">• Melakukan <i>code refactoring</i> terhadap fitur <i>update profile</i> pada BIGRR (<i>tools</i> perusahaan)• Mengerjakan <i>Onboarding assignment</i> dan melakukan presentasi pengetahuan mengenai perusahaan kepada HR Team
2	<ul style="list-style-type: none">• Melakukan <i>testing</i> pada hasil <i>refactor</i>• Melakukan <i>code refactoring</i> dan <i>feature improvement</i> terhadap fitur <i>Property Notes</i> pada BIGRR• Melakukan perubahan beberapa komponen UI (<i>User Interface</i>) pada kode BIGRR menjadi <i>Global Component</i>
3	<ul style="list-style-type: none">• Melakukan <i>code refactoring</i> dan <i>feature improvement</i> terhadap fitur <i>Property Notes</i> pada BIGRR• Melakukan <i>testing</i> pada hasil <i>refactor</i> dan <i>improvement</i> fitur <i>Property Notes</i>
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
4	<ul style="list-style-type: none"> • Melakukan <i>briefing</i> untuk proyek Promenade • Mempelajari penggunaan Plotly Dash dan Python untuk pengembangan proyek Promenade • Melakukan pembuatan <i>UI Components</i> untuk website Promenade menggunakan Plotly Dash
5	<ul style="list-style-type: none"> • Membuat fitur <i>Collect Data</i> dan <i>Push Rates to Beds24</i> • Membuat komponen Dropdown <i>reusable</i> dan <i>date picker</i> • Membuat komponen tabel untuk menampilkan data <i>Pricing Decision</i> • Membuat <i>loading indicator</i> pada tabel data • Membuat filter untuk data didalam tabel berdasarkan <i>start date</i> dan <i>end date</i>
6	<ul style="list-style-type: none"> • Membuat fitur <i>staged, modified</i> pada tabel data • Melakukan <i>refactoring</i> pada kode dan komponen Promenade • Membuat fitur <i>fetching</i> data <i>Pricing Decision</i> dari <i>Backend</i> ke dalam tabel • Membuat laporan magang
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
7	<ul style="list-style-type: none"> • Memperbaiki tampilan secara keseluruhan • Mengimplementasikan <i>user validation</i> ketika <i>login</i> ke dasbor • Melakukan <i>refactoring</i> pada kode dan komponen Promenade • Melakukan <i>setup</i> untuk proyek Gaia-Interface • Menambahkan <i>button</i> "<i>Reply Manually</i>" pada fitur "<i>Auto Reply</i>" Gaia-Interface • Membuat laporan magang
8	<ul style="list-style-type: none"> • Membuat fitur BV-GO Message <i>slash command</i> • Melakukan <i>debugging</i> fitur <i>slash command</i> • Melakukan <i>bug fixing</i> pada halaman Booking Calendar website BIGRR • Membuat laporan magang
9	<ul style="list-style-type: none"> • Melakukan <i>bug fixing</i> pada halaman Property website BIGRR • Membuat fitur "<i>Add Integration</i>" pada Gaia-Interface • Melakukan <i>testing</i> dan <i>debugging</i> fitur "<i>Add Integration</i>" • Membuat laporan magang
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
10	<ul style="list-style-type: none"> • Memperbaiki halaman Newsletter pada website BIGRR • Membuat fitur <i>search</i> dan <i>filter</i> pada halaman Newsletter • Melakukan <i>refactoring</i> kode Newsletter BIGRR • Membuat laporan magang
11	<ul style="list-style-type: none"> • Memperbaiki fitur <i>insert</i> dan <i>update</i> Newsletter • Melakukan <i>refactoring</i> kode Newsletter BIGRR • Melakukan <i>folder restructuring</i> pada proyek Newsletter BIGRR • Melakukan <i>testing</i> dan <i>debugging</i> fitur pada halaman Newsletter • Membuat laporan magang
12	<ul style="list-style-type: none"> • Membuat fitur <i>Opt-Out Specific Guest</i> pada Gaia-Interface • Membuat laporan magang
Lanjut pada halaman berikutnya	

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
13	<ul style="list-style-type: none"> • Melanjutkan pembuatan fitur <i>Opt-Out Specific Guest</i> pada Gaia-Interface • Melakukan <i>testing</i> dan <i>debugging</i> fitur <i>Opt-Out Specific Guest</i> • Melakukan <i>monitoring</i> fitur <i>Opt-Out Specific Guest</i> • Membuat laporan magang

- Minggu ke-1 sampai ke-3

Pada minggu pertama hingga minggu ketiga, pekerjaan yang dilakukan kebanyakan berfokus pada pengembangan beberapa fitur dan melakukan *refactoring* kode dari website BIGRR (*tools* internal). Minggu pertama diberikan tugas untuk melakukan *refactoring* kode pada fitur *update profile* yang merupakan salah satu fitur pada BIGRR. *Refactoring* ini bertujuan untuk menyesuaikan dengan *design pattern* yang baru dan membuat kode menjadi lebih rapih sehingga mempermudah *developer* ketika melakukan *development* maupun *maintenance*. Durasi pengerjaan memakan waktu hampir dua minggu karena memerlukan pemahaman kode terlebih dahulu dan mengerti alur dan kegunaan setiap *functions* yang ada didalam kode.

Setelah dilakukan pengembangan dan *refactoring*, selanjutnya dilakukan *testing*. *Testing* dilakukan untuk memastikan fitur dan perubahan kode-kode yang dibuat dapat berjalan dengan baik sebelum di *deploy* ke *branch* master nantinya. *Testing* juga dilakukan untuk meminimalisir terjadinya *bug* dan *error* ketika digunakan oleh *user*.

Selain mengerjakan *refactoring*, terdapat *onboarding assignment* yang diberikan kepada setiap pekerja magang yang bertujuan untuk lebih memahami perusahaan dan *core business* perusahaan itu sendiri. Selain itu, *assignment* tersebut juga berguna untuk memahami teknologi-teknologi apa saja yang digunakan pada setiap *tools* internal yang ada. Setelah mengerjakan *onboarding assingment*, setiap pekerja magang akan diminta

untuk mempresentasikan apa yang telah dipelajari dan diketahui mengenai perusahaan dan teknologi-teknologi yang digunakan, khususnya pada divisi *Engineering*.

Setelah mengembangkan dan *refactoring* fitur *update profile*, pada minggu yang sama juga melakukan pengembangan dan *refactoring* fitur lain pada BIGRR, yaitu *property notes*. Fitur ini merupakan fitur yang berguna untuk memberikan catatan pada setiap *property* milik perusahaan. Pengembangan ini menghabiskan waktu satu minggu dan sudah bersamaan dengan *testing*. Selama proses tersebut, dilakukan juga implementasi *Global Component* pada kode milik fitur ini.

- Minggu ke-4 sampai ke-7

Pada minggu ke empat hingga minggu ke tujuh, berfokus pada pembuatan website dasbor "Promenade" yang mana merupakan *tools* internal baru milik perusahaan. Pada minggu ke empat, dilakukan *briefing* terlebih dahulu mengenai proyek yang akan dikerjakan. Lalu setelah mengetahui apa saja yang diperlukan untuk melakukan *development*, dilakukan pendalaman cara penggunaan teknologi-teknologi dan mulai membuat beberapa komponen *User Interface* menggunakan Plotly Dash.

Pada minggu ke lima hingga ke enam, dimulai pembuatan beberapa fitur utama, yaitu fitur *Collect Data* yang berguna untuk mengambil data *pricing decision* dari database milik *user* dengan *range* tanggal yang sudah ditentukan dan data tersebut akan ditampilkan kedalam tabel. Setelah itu, dibuat juga beberapa fitur tambahan, seperti fitur filter data berdasarkan *start date* dan *end date*, fitur *staged modified* pada tabel yang berguna untuk memberikan indikasi ketika data pada tabel diubah (*modified*) dan ketika perubahan sudah disimpan (*staged*), dan fitur *Push Rates to Beds24* yang akan mengirimkan data *pricing decision* pada tabel ke Beds24 milik perusahaan.

Pada minggu ke tujuh, pengerjaan proyek Promenade sudah memasuki tahap akhir, dan tersisa perbaikan beberapa tampilan dan melakukan *refactoring* kode beberapa komponen *User Interface* dan membuat *user validation* untuk memastikan *user* yang mengakses halaman dasbor merupakan *user* yang sudah *login*.

Setelah proyek Promenade selesai, diberikan tugas baru lagi pada *tools* yang berbeda, yaitu Slack Bot yang bernama GAIA. Tugas pertama yang

diberikan adalah untuk membuat fitur *Reply Manually* pada GAIA. Fitur ini berguna bagi *user* untuk membalas pesan dari *guest* secara manual jika pesan tersebut dianggap melebihi kapasitas GAIA dalam menjawab pesan.

- Minggu ke-8

Pada minggu ke delapan, diberikan tugas baru untuk membuat fitur baru pada GAIA, yaitu fitur Slash Command BV-GO Message yang dapat digunakan ketika *user* mengetikkan `"/bv-go-message"` maka langsung memunculkan *modal box* yang berisikan *input field* untuk mengirimkan pesan ke *partner*. Setelah menyelesaikan fitur tersebut, dilakukan *bug fixing* terhadap fitur *Booking Calendar*.

- Minggu ke-9

Pada minggu ke sembilan, melakukan *bug fixing* pada halaman Property BIGRR. Setelah itu, membuat fitur baru pada GAIA, yaitu "Add Integration" yang berguna untuk *user* jika ingin menambahkan Slack Bot baru pada proyek GAIA.

- Minggu ke-10 sampai ke-11

Pada minggu ke sepuluh hingga sebelas, dilakukan perbaikan dan pengembangan fitur Newsletter pada BIGRR dengan menambahkan fitur *search* dan *filter* dan juga melakukan *refactoring* kode Newsletter sehingga mengikuti *design pattern* yang terbaru. Setelah itu, dilakukan perbaikan pada fitur *insert* dan *update newsletter* dan juga *refactoring* seluruh kode dan *folder restructuring* milik fitur Newsletter.

- Minggu ke-12 sampai ke-13

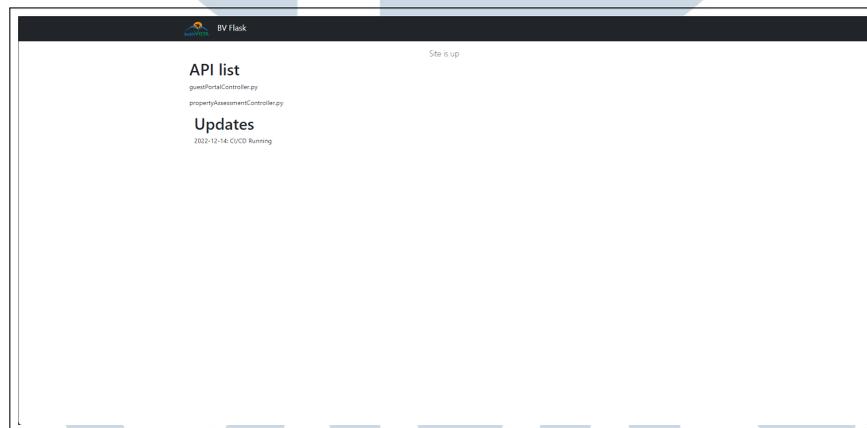
Pada pekan kedua belas hingga tiga belas magang, fokus kembali diarahkan pada pengembangan GAIA dengan penambahan fitur baru, yakni *Opt-Out Specific Guest*. Fitur ini memiliki tujuan penting, yaitu mencegah GAIA untuk secara otomatis membalas pesan dari tamu tertentu. Pembuatan fitur ini merespon keluhan yang cukup signifikan dari para tamu, terutama terkait dengan kebingungan saat mengetahui bahwa respons atas pesan mereka berasal dari GAIA. Keadaan ini menjadi lebih krusial saat terdapat kasus-kasus di mana tamu tengah menyampaikan keluhan, dan GAIA memberikan respons, sedangkan seharusnya tanggapan tersebut diserahkan kepada tim layanan pelanggan. Hal ini menggarisbawahi upaya kami untuk

memastikan pengalaman komunikasi yang lebih efektif dan meminimalkan potensi kebingungan bagi para tamu.

3.4 Proses Pelaksanaan Pengerjaan Proyek

3.4.1 Pembuatan *NavBar* Dasbor Promenade

Setelah tiga minggu menjalani magang di perusahaan Bukit Vista dan berfokus pada pengembangan beberapa fitur dan *refactoring* website BIGRR, diberikanlah proyek baru dengan teknologi yang berbeda dari yang digunakan pada saat pengembangan website BIGRR. Proyek ini merupakan pembuatan dasbor yang berguna untuk melihat perkembangan perusahaan dengan beberapa statistik yang tervisualisasikan dalam bentuk grafik interaktif dan juga beberapa fitur untuk melakukan *dynamic pricing*. Pembuatan dasbor ini dibuat didalam website Flask Bukit Vista hanya saja memiliki halamannya tersendiri dan membutuhkan autentikasi untuk dapat mengakses halaman dasbor ini.



Gambar 3.1. Home Page Flask Bukit Vista Website

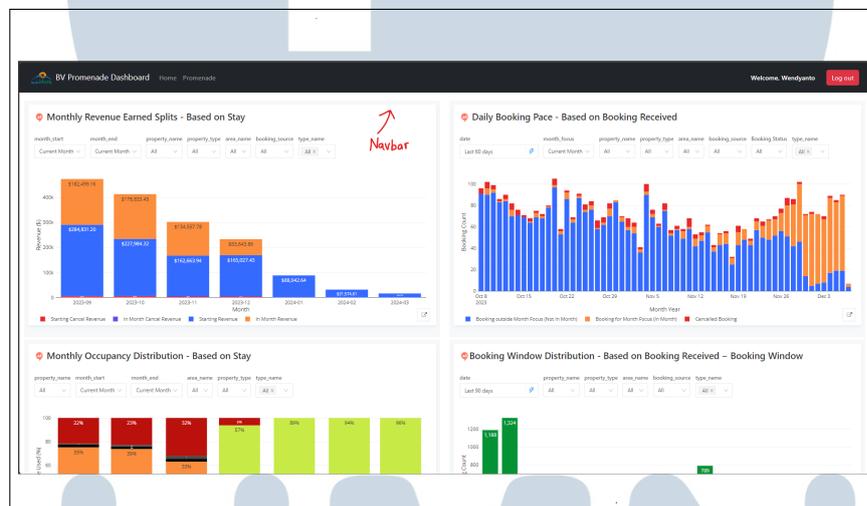
Pada proses pembuatan dasbor Promenade ini, komponen pertama yang dibuat adalah *NavBar* atau *Navigation Bar* yang berguna untuk menempatkan beberapa menu navigasi yang dapat digunakan oleh *user* nantinya. *NavBar* pada website Promenade terdiri dari beberapa *item*, antara lain:

- Logo perusahaan Bukit Vista
- Navigasi "Home"
- Navigasi "Promenade"

- Nama *user*
- Navigasi *Login* atau *Logout*.

Navigasi "Home" berguna untuk memindahkan *user* ke halaman awal website Flask Bukit Vista dengan tampilan seperti pada gambar 3.1. Sedangkan navigasi "Promenade" berguna untuk memindahkan *user* ke halaman utama dasbor Promenade yang tampilannya seperti pada gambar 3.2.

Terdapat teks yang bertuliskan nama *user* yang mengakses dasbor tersebut dan setelah nama tersebut terdapat tombol *logout* yang berfungsi untuk *user* keluar dari halaman tersebut jika sudah tidak ingin mengakses halaman dasbor Promenade.

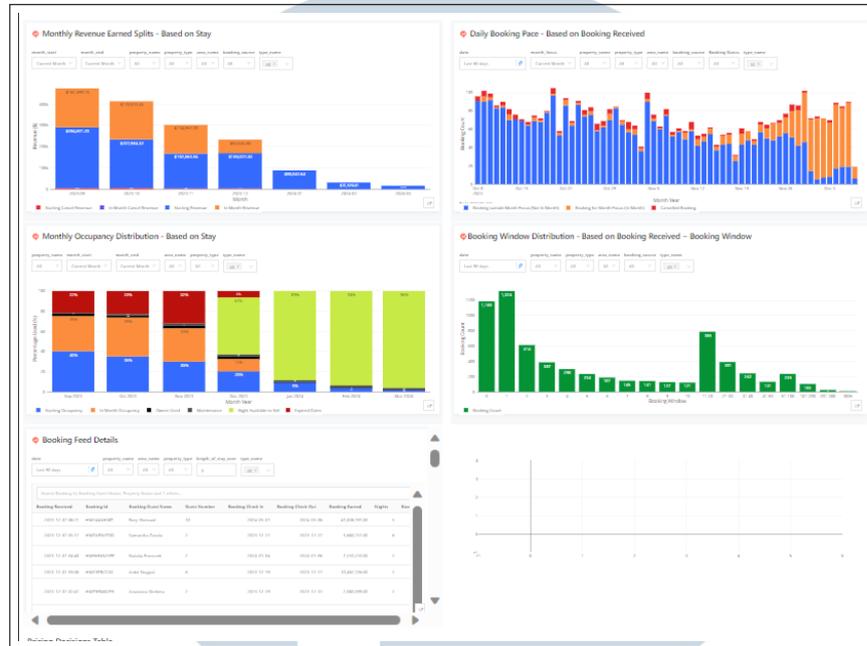


Gambar 3.2. Navigation Bar Promenade

Pada gambar 3.2, telah ditandai bagian NavBar yang berada pada posisi paling atas dasbor dan berwarna hitam. Semua navigasi yang disebutkan sebelumnya juga dapat dilihat pada gambar tersebut.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.4.2 Pengerjaan *Embedding* Beberapa Visualisasi dari DataVista



Gambar 3.3. *Embedded Visualizations*

Pada Gambar 3.3, terlihat sejumlah grafik pada dasbor ini yang mencerminkan statistik hasil *embedding* dari *tools* eksternal yang bernama DataVista. Berikut merupakan penjelasan detail mengenai setiap grafik yang ditampilkan.

- *Monthly Revenue Earned Splits*

Grafik *Monthly Revenue Earned Splits* berguna untuk melihat performa properti yang akan dilakukan *pricing decision* oleh Revenue Manager sehingga relatif dengan performa pada bulan-bulan sebelumnya.

- *Daily Booking Pace*

Grafik *Daily Booking Pace* digunakan untuk melihat perlajuan jumlah *booking* pada semua properti setiap harinya dan berguna untuk melihat momentum dari properti yang ingin dilakukan *pricing* ulang.

- *Monthly Occupancy Distribution*

Grafik *Monthly Occupancy Distribution* ini berguna untuk mengetahui perkembangan *revenue* pada bulan lalu. Contohnya, apabila terjadi penurunan

revenue pada bulan lalu, dapat dianalisa penyebab terjadinya penurunan tersebut.

- *Booking Window Distribution*

Grafik *Booking Window Distribution* digunakan untuk mengetahui apakah jarak *booking* suatu properti cenderung dilakukan pada saat *last minute* atau jauh, seperti 1-2 minggu atau lebih.

- *Table Explainer*

Grafik ini terletak dibawah grafik *Booking Window Distribution* yang mana grafik ini berguna untuk memvisualisasikan data dari tabel *Pricing Decision* dan data yang divisualisasikan adalah data yang berada pada baris tabel yang di *click* oleh *user*. Sayangnya, implementasi dari grafik ini tidak dilanjutkan sehingga pada saat dasbor ini selesai dibuat, grafik tersebut masih belum bisa digunakan.



```

dbc.Row(
  [
    dbc.Col(
      html.Iframe(
        src="https://datavista.bukitvista.com/embed/query/728/visualization/1492?
        api_key=oDu7Goj5b5oQMg20dXPEmb9vLmJU5NqS3S2XKINI&p_month_start=Current%20Month&
        p_month_end=Current%20Month&p_property_name=All&p_property_type=All&
        p_area_name=All&p_booking_source=All&p_type_name=%5B%22All%22%5D", Line too lon
        width="100%",
        height=590,
      )
    ),
    dbc.Col(
      html.Iframe(
        src="https://datavista.bukitvista.com/embed/query/84/visualization/535?
        api_key=pxabMcrhfC4VmAk7LyD1cORjHNJgDOvpMwrgP72R&p_date=d_last_60_days&
        p_month_focus=Current%20Month&p_property_name=All&p_property_type=All&
        p_area_name=All&p_booking_source=All&p_booking_status=All&
        p_type_name=%5B%22All%22%5D", Line too long (335/100)
        width="100%",
        height=590,
      )
    ),
  ],
  style={"margin-top": 20},
),
dbc.Row(
  [
    dbc.Col(
      html.Iframe(
        src="https://datavista.bukitvista.com/embed/query/744/visualization/1518?
        api_key=LvMMUzzy20IHcStNifSIOG0aQPkLcdtFGVGhppqJ&p_property_name=All&
        p_month_start=Current%20Month&p_month_end=Current%20Month&p_area_name=All&
        p_property_type=All&p_type_name=%5B%22All%22%5D", Line too long (301/100)
        width="100%",
        height=590,
      )
    ),
    dbc.Col(
      html.Iframe(
        src="https://datavista.bukitvista.com/embed/query/101/visualization/1451?
        api_key=lyJLM3oUDQifTLrm4f36KwK33TcZDkexq0YVUVVic&p_date=d_last_90_days&
        p_property_name=All&p_property_type=All&p_area_name=All&p_booking_source=All&
        p_type_name=%5B%22All%22%5D", Line too long (286/100)
        width="100%",
        height=590,
      )
    ),
  ],
),
),

```

Gambar 3.4. Potongan kode untuk melakukan *embedding* grafik DataVista

Pada gambar 3.4 merupakan potongan kode yang berguna untuk menampilkan beberapa grafik seperti pada gambar 3.3. Untuk menampilkan grafik tersebut digunakan sintaks "html.Iframe" yang sudah disediakan oleh Plotly Dash dan meletakkan *link* sumber dari grafik yang ingin ditampilkan pada atribut "src".

Proses *embedding* ini dilakukan dengan tujuan menghindari duplikasi pembuatan beberapa grafik interaktif yang telah disajikan oleh DataVista. Hanya beberapa visualisasi yang terkait dengan dasbor ini yang diambil dari DataVista, seperti perkembangan *revenue* dan *booking pace*, yang memiliki relevansi langsung dengan fungsi dasbor ini. *Embedding* ini juga dilakukan dengan maksud agar *user* tidak perlu mengakses DataVista secara terpisah untuk melihat grafik-grafik tertentu yang berhubungan dengan pengambilan keputusan terkait *pricing decision*.

3.4.3 Pengerjaan Fitur *Staged Pricing Decision*

Room_id	Unit Name	Inventory	Max_capacity	Is_multi_unit	Listing_id	Db_property_id	Available_units	Start Date	End Date	Property_id
431186	BOX A & B 10 PAX	2	10	1		PR0782	2	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0782
442255	Dorm (Bunk Bed) Couple	4	2	0	974264859799497762	PR0782	4	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0782
452924	Dorm (Bunk Bed)s Entire	1	8	1	93364833960283584	PR0782	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0782
398514	2BR Double	1	4	1	24572193	PR0126	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0126
387365	Entire	1	8	0	597359833771691214	PR0707	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0707
387372	Entire	1	6	0	71866598033426149	PR0755	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0755
387369	Entire	1	6	0	76106741773025297	PR0720	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0720
397680	Twin	1	2	0	96360965	PR0442	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0442
398523	2BR	1	4	1	20445152	PR0198	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0198
399517	Wooden Twin	3	2	0	23968183	PR0198	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0198
387366	Entire	1	8	0	30276593	PR0355	1	2023-09-19T00:00:00	2023-09-21T00:00:00	PR0355

Gambar 3.5. *Pricing Decision table* sebelum *user* melakukan perubahan data

id3	Price_max	Limit Rate	Min_stay	Min_stay_flexibility	Min_advance	End Offset	Length	Fill_rate	Room_ratio	Sped	Staged
0	150000	1	0	72	54.480000000000004	1	3	0	1	0	1
00	310000	320000	1	0	72	2.25	1	-3.036	0		
000	1160000	1520000	1	0	72	9	1	-18.542630158730163	0		
00	650000	650000	1	0	72	26.04	1	-0.8773888888888893	0		
000	2020000	1500000	1	0	72	0	1	3	1		
000	1410000	900000	1	0	72	1	1	-3.2355555555555555	0		
000	1890000	850000	1	0	3	17.01	1	-2.4887777777777775	0		
00	550000	440000	1	0	3	7.02	1	-5.255000000000002	0		
00	1190000	800000	1	0	3	108.99000000000001	1	-1.8175555555555556	0		
00	500000	450000	1	0	12	3.66	0.3333333333333333	-2.0736111111111111	0		
000	1040000	750000	1	0	9	2	1	-3.7594047619047624	0		

Gambar 3.6. *Pricing Decision table* setelah *user* melakukan perubahan data

Fitur *staged* memiliki peran signifikan dalam memberikan indikasi pada baris tabel yang telah mengalami perubahan, memungkinkan *user* untuk dengan mudah melacak data yang baru saja diubah. Ketika *user* melakukan modifikasi, sebuah *button staging* akan muncul, memfasilitasi *user* untuk melakukan *staging* terhadap perubahan yang telah dilakukannya sebelumnya. Fitur ini menjadi sangat relevan karena mengatasi tantangan di mana *user* mungkin lupa terhadap perubahan yang telah dilakukannya jika tiba-tiba meninggalkan dasbor dan kembali. Dengan adanya fitur ini, *user* tidak perlu khawatir kehilangan jejak terhadap perubahan data.

Ilustrasi pada Gambar 3.5 menunjukkan tampilan tabel dalam keadaan normal tanpa adanya perubahan yang dilakukan oleh *user*. Namun, seiring dengan modifikasi yang dilakukan oleh *user*, baris yang terkena perubahan akan berubah warna menjadi kuning dan *button staging* akan muncul dibawah elemen *date picker*

yang berguna untuk menyimpan perubahan yang telah diubah oleh *user*, seperti yang diperlihatkan pada Gambar 3.6.

```

if column_definitions is None:
    column_definitions = [{"field": col_name} for col_name in df.columns]

    if editable_col is not None:
        column_definitions = [
            {
                "field": col_name,
                "editable": True if col_name in editable_col else False,
            }
            for col_name in df.columns
        ]

row_style = {
    "styleConditions": [
        {
            "condition": "params.data.status == 1",
            "style": {"backgroundColor": "rgb(253 224 71)"},
        },
        {
            "condition": "params.data.status == 2",
            "style": {"backgroundColor": "rgb(163 230 53)"},
        },
    ]
}

```

Gambar 3.7. Potongan kode untuk mengganti *background color* dari *row* pada tabel

Pada gambar 3.7 merupakan kode yang berguna untuk mengganti *background color* milik baris yang datanya di ganti oleh *user*. Perubahan dilakukan secara kondisional menggunakan sintaks *”if”*. Terdapat *object* yang memiliki properti bernama *styleCondition* yang juga menyimpan *array* berisikan *object* yang memuat kondisi-kondisi kapan *background color* dari suatu baris harus berubah dan juga warna apa yang harus digunakan.

3.4.4 Pengerjaan Fitur *Filter Pricing Decision by Property*

IDENTIFIER COLUMNS		ESSENTIAL INFORMATION				EDITABLE COLUMNS				AUTO GENERATED			
date executed	area name	property name	room name	start date	end date	minimum rate	current price	suggest... price change	suggest... price	price change	weighted effort	min stay	final price
2023-10-03	Nusa Perida	Acala Host...	Dormitory	2023-10-13	2023-10-14	150000	3020000	-12.468	2540000			1	
2023-10-03	Nusa Perida	Acala Host...	3BR	2023-10-08	2023-10-14	0	0	0.780556	0			1	
2023-10-03	Nusa Perida	Acala Host...	3BR	2023-10-15	2023-10-17	0	0	0.514167	0			1	
2023-10-03	Nusa Perida	Acala Host...	3BR	2023-10-06	2023-10-07	0	0	0.0102778	0			1	
2023-10-03	Nusa Perida	Acala Host...	3BR	2023-10-03	2023-10-05	0	0	0.131111	0	30		1	0
2023-10-03	Nusa Perida	Acala Host...	Dormitory	2023-10-13	2023-10-17	150000	1480000	-12.3718	1300000			1	

Gambar 3.8. Hasil *filter by property*

Fitur yang diusulkan merupakan sebuah tambahan yang sangat bermanfaat bagi pengguna untuk menampilkan dan mengubah data properti tertentu. Keberadaan fitur ini menjadi sangat penting mengingat jumlah data properti pada tabel yang tersedia dapat mencapai ribuan, menciptakan kebutuhan untuk melakukan penyaringan secara efektif. Dalam implementasinya, fitur ini, dikenal sebagai *Filter by Property*, memungkinkan pengguna untuk menyaring data berdasarkan properti tertentu, dan hasilnya tergambar pada Gambar 3.8. Proses penyaringan dilakukan dengan memanfaatkan *property_id* yang dimiliki oleh setiap properti, yang tentunya bersifat unik dan berbeda satu sama lain. Pendekatan ini mempermudah proses *filtering* data pada tabel

The screenshot shows a web interface titled "Modified Pricing Decisions Table". At the top, there is a "Filter by" dropdown menu with "Property" selected and "Villa Melasti Ungasan" chosen. Below the filter is a table with the following columns:

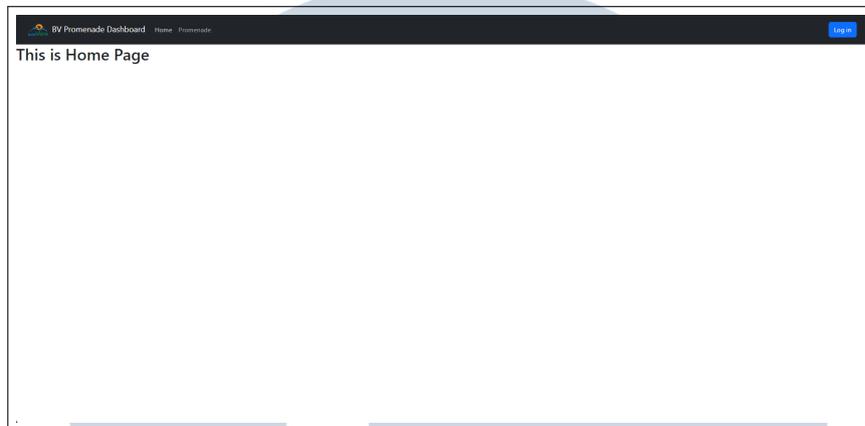
IDENTIFIER COLUMNS						ESSENTIAL INFORMATION				EDITABLE COLUMNS			AUTO GEN...
date executed	area name	property name	room name	start date	end date	minimum rate	current price	suggest... price change	suggest... price	price change	weekend offset	min stay	final price
No Rows To Show													

At the bottom right of the table, there is a pagination control showing "0 to 0 of 0" and "Page 0 of 0".

Gambar 3.9. Hasil ketika properti tidak ditemukan pada tabel data

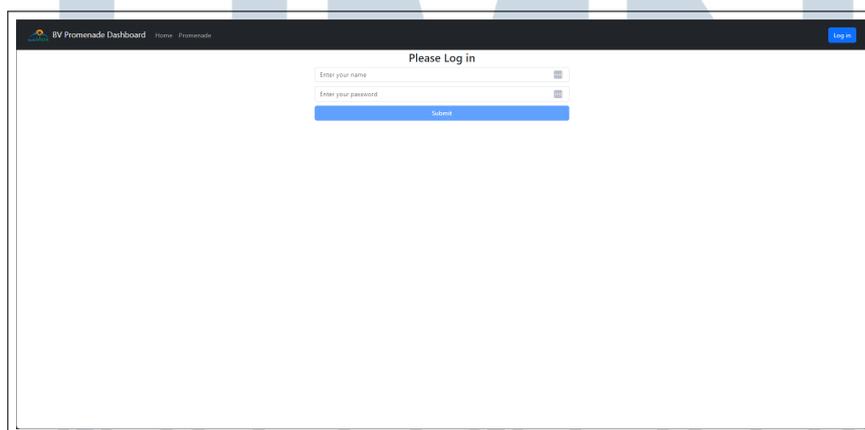
Pada ilustrasi gambar 3.9, terlihat bahwa tidak ada data yang tampak. Kondisi ini muncul karena tabel tidak memuat data untuk properti yang telah dipilih, sehingga tidak ada data yang ditampilkan pada tabel, melainkan hanya pesan teks informatif berupa "No rows to show". Situasi ini menandakan bahwa tidak ada catatan yang memenuhi kriteria filtrasi tertentu pada properti yang sedang ditinjau pada saat tersebut.

3.4.5 Implementasi Autentikasi *User*



Gambar 3.10. Halaman *Home* milik dasbor Promenade

Untuk mencegah akses yang tidak sah ke dasbor, autentikasi telah diimplementasikan untuk setiap pengguna yang berupaya mengakses Dasbor Promenade. Pengguna diharuskan melakukan *login* melalui halaman *login* yang dapat diakses dengan menekan tombol *login* yang terdapat pada *NavBar*, sebagaimana terlihat pada Gambar 3.10. Dengan langkah-langkah ini, akses dasbor hanya dapat diberikan kepada pengguna yang telah melakukan otentikasi, sehingga menjaga keamanan dan menghindari akses yang tidak diizinkan.



Gambar 3.11. Halaman *Login* milik dasbor Promenade

Gambar 3.11 menggambarkan halaman *login*. Ketika *user* telah memilih opsi *login* dari halaman *home* sebelumnya, sistem akan mengarahkannya ke halaman ini. Akses ke dasbor Promenade hanya diperbolehkan untuk *user*

yang menggunakan akun perusahaan terdaftar dalam *database*. Pembatasan ini diberlakukan untuk memastikan bahwa hanya pihak internal perusahaan yang berhak mengakses dasbor ini, mengingat data yang terdapat di dalamnya bersifat sensitif. Jika ada *user* yang mencoba masuk ke halaman dasbor tanpa melakukan proses *login* sebelumnya, sistem akan terus mengarahkannya ke halaman *login* untuk memastikan keamanan akses.

3.4.6 Penyempurnaan *User Interface* dan *Experience* Pada Tabel *Pricing Decision*

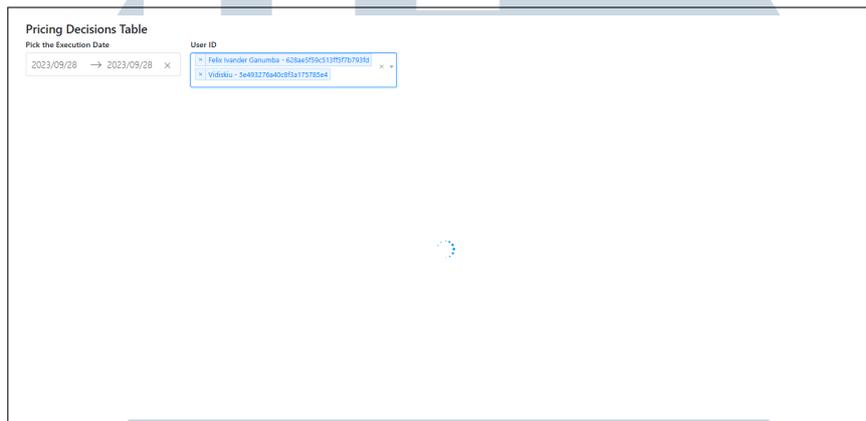
Dalam fase akhir setelah semua fitur telah diimplementasikan dan dapat digunakan, langkah selanjutnya adalah menyempurnakan tampilan beberapa komponen yang masih memungkinkan untuk ditingkatkan. Beberapa penyesuaian dilakukan khususnya pada aspek tabel *pricing decision*. Peningkatan ini merespons umpan balik dari beberapa pengguna dengan tujuan meningkatkan pengalaman pengguna (*user experience*) serta memperindah tampilan dasbor, terutama pada bagian tabel *pricing decision* yang merupakan komponen yang paling sering diakses. Proses peningkatan ini bertujuan untuk memberikan antarmuka yang lebih baik dan responsif, memenuhi harapan pengguna, dan meningkatkan estetika tampilan keseluruhan dari dasbor.

Pricing Decisions Table													
Show Data by		Execution Date		User ID									
2023/10/11 → 2023/10/11		2023/10/11		Felix Ivander Ganumba		Stage Changes							
Filter by Property													
All													
All Data						Staged			Pushed				
IDENTIFIER_COLUMNS						ESSENTIAL INFORMATION			EDITABLE COLUMNS				AUTO GENL.
date executed	area name	property name	room name	start date	end date	minimum rate	current price	suggest. price change	suggest. price	price change	weekend offset	min stay	final price
2023-10-11	Canggu	Armadillo ...	Armadillo ...	2023-10-16	2023-10-20	1,400,000	1,240,000	-10,0558	1,120,000	10,000		1	125,240,000
2023-10-11	Canggu	Armadillo ...	Double	2023-10-16	2023-10-20	360,000	360,000	-2,25944	350,000	0		1	0
2023-10-11	Canggu	Armadillo ...	New Build ...	2023-10-22	2023-10-23	325,000	420,000	-3,93437	400,000	0		1	0
2023-10-11	Nusa Peni ...	Ba Bar Cot ...	3BR Bunga ...	2023-10-13	2023-10-13	720,000	890,000	-8,91794	810,000	0		1	0
2023-10-11	Nusa Peni ...	Ba Bar Cot ...	Bungalow ...	2023-10-13	2023-10-13	960,000	1,140,000	-8,7845	1,040,000	0		1	0
2023-10-11	Padang Pa ...	Despacito ...	Despacito ...	2023-10-29	2023-11-04	710,000	970,000	-2,81272	940,000	0		1	0
2023-10-11	Padang Pa ...	Despacito ...	Despacito ...	2023-11-05	2023-11-11	710,000	940,000	-1,68032	880,000	0		1	0
2023-10-11	Padang Pa ...	Despacito ...	Despacito ...	2023-10-29	2023-11-04	650,000	940,000	-1,32009	890,000	0		1	0

Gambar 3.12. Tabel *pricing decision*

Dimulai dengan melakukan perbaikan pada jenis tabel yang sebelumnya menampilkan *pricing decision*, data *staging*, dan data *pushed* dalam tabel yang terpisah, mengakibatkan tata letak dasbor terlihat panjang secara vertikal. Sebagai solusi, tabel *pricing decision* telah dimodifikasi untuk diintegrasikan dalam

satu tabel, namun dengan setiap kategori data ditempatkan pada *tab* terpisah, sebagaimana terillustrasi pada gambar 3.12. Pembaruan ini bertujuan untuk meningkatkan *user experience* dengan mengeliminasi kebutuhan *scrolling* vertikal untuk melihat data *staging*, *pushed*, dan memungkinkan pengguna hanya perlu beralih antar *tab* untuk mengakses informasi yang diinginkan.



Gambar 3.13. Loading indicator pada tabel *pricing decision*

Setelahnya, penyempurnaan terakhir diterapkan pada tabel *pricing decision* dengan penambahan indikator pemuatan (*loading indicator*), seperti yang terlihat pada gambar 3.13. Indikator pemuatan ini akan tampil saat data keputusan harga sedang diambil dari backend.



Gambar 3.14. Potongan kode untuk menampilkan *loading indicator*

Pada gambar 3.14 merupakan kode yang digunakan untuk memunculkan *loading indicator* ketika memuat data *pricing decision* sebelum ditampilkan pada tabel nantinya. Plotly Dash menyediakan komponen *loading* yang dikategorikan sebagai Dash Core Component yang merupakan salah satu *module* yang disediakan

oleh Plotly Dash yang memungkinkan untuk menggunakan berbagai komponen interaktif yang sudah disediakan contohnya komponen *loading indicator* ini.

Keberadaan indikator pemuatan ini memberikan informasi kepada pengguna bahwa sistem sedang memperoleh data yang diperlukan. Tanpa adanya indikator ini, pengguna mungkin akan merasa bingung karena tidak ada respons saat pengambilan data, dan mereka dapat menganggap bahwa dasbor Promenade mengalami masalah.

3.5 Perangkat dalam Pengembangan Proyek

Dalam proses pengembangan proyek yang dikerjakan selama masa magang berlangsung, terdapat beberapa perangkat yang digunakan untuk menunjang pengerjaan proyek. Berikut perangkat-perangkat yang digunakan.

3.5.1 Perangkat Keras

- Legion Y540-15IRH-PG0 (Laptop)
- NVIDIA GEFORCE GTX 1650 (GPU)
- Intel Core i7-9750HF @2.60GHz (CPU)

3.5.2 Perangkat Lunak

- Windows 11 Home (*operating system*)
- Visual Studio Code (*code editor*)
- Python (bahasa pemrograman)
- Plotly Dash (*UI components library*)
- Pandas (*data manipulation library*)
- Flask (*framework*)
- Microsoft Edge (*browser*)
- Git (*version control*)
- GitHub (*version control platform*)

3.6 Kendala dan Solusi yang Ditemukan

3.6.1 Kendala yang ditemukan

Berikut merupakan kendala-kendala yang dihadapi selama proses pembuatan dasbor Promenade ini.

1. Ketika melakukan *refactoring* kode untuk membuat kode untuk komponen *user interface* dan kode untuk fungsionalitas tidak pada satu *file* yang sama karena dapat membuat kode menjadi *spaghetti code* dan sulit untuk melakukan *maintenance* nantinya.
2. Kesulitan saat melakukan *data handling* menggunakan *library* Pandas pada bahasa pemrograman Python dan ketika membuat beberapa fungsi interaksi pada komponen dasbor seperti *dropdown*, *table*, *input field*.

3.6.2 Solusi yang diterapkan

Adapun solusi yang diterapkan untuk mengatasi kendala-kendala yang disebutkan sebelumnya.

1. Mengalokasikan waktu untuk melakukan *trial and error* hingga menemukan cara untuk membuat proyek menjadi *modularize* dengan memisahkan kode komponen *user interface* dan kode untuk fungsionalitas.
2. Membaca dan menelusuri dokumentasi dari teknologi-teknologi tersebut karena terdapat penjelasan dan contoh penggunaannya. Sesekali mencari melalui *Stack Overflow* dan bertanya langsung kepada *supervisor* proyek.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A