

ECOMMERCE DASHBOARD - PAPUA TEAM

Hendra Yoga Wiguna

Kevin Reinhard Ambarita

Billy Samosir

A. Problem Definition

Perusahaan sebagai organisasi profit didirikan untuk menghasilkan keuntungan dan juga mencapai tujuannya masing-masing. Banyak faktor yang mempengaruhi sebuah perusahaan untuk bisa mempertahankan *sustainability*, terutama perusahaan yang berbasis teknologi/*start-up*. Sebagai perusahaan *e-commerce* baru yang banyak menghasilkan dan mengutilisasi data, banyak hal yang perlu dipertimbangkan oleh *stakeholders*, mulai dari *upper*, *middle* hingga *lower management* untuk melakukan *tracking* terhadap *metrics*, *performance* dan *daily operational*.

Kebutuhan masing-masing *stakeholders* juga berbeda. Misalnya *upper management* yang terdiri dari C-Level membutuhkan data yang mendukung mereka dalam menyusun strategi yang bersifat *mid to long term planning*. Karena waktu mereka juga sangat terbatas akibat kesibukan lain yang perlu dijalankan, diperlukan *data visibility* yang tinggi dan *insightful* bagi mereka. Team *sales* dan *operational* mungkin memerlukan perencanaan yang lebih taktikal dibanding strategis, karena mereka berurusan langsung dengan lapangan. Untuk itulah, diperlukan *metrics* yang lebih detail supaya bisa menunjukkan *sense of urgency* atas kondisi nyata. Team *marketing* memerlukan suatu wadah untuk memastikan “*the right product at the right place (person)*” sehingga mereka perlu mengkaji *initiatives* yang telah dilakukan melalui *campaign* dan juga segmentasi *customer* untuk membedakan strategi yang ingin diterapkan berdasarkan karakteristik masing-masing *customer*. *Business development* memerlukan *insight* apakah *initiative* yang mereka lakukan cukup *feasible* untuk diterapkan di perusahaan sehingga diperlukan kesimpulan ringkas atas eksperimen yang dilakukan.

Berangkat dari masalah-masalah ini, diperlukan suatu produk berupa *dashboard* yang harapannya bisa menjawab masalah-masalah yang dihadapi para *user*.

User Definition

1. CEO

Sebagai sarana untuk melihat performa perusahaan secara utuh dan menentukan rencana taktikal dan strategis jangka pendek, menengah dan panjang. *Chief Operating Officer* (CEO) dapat mengkaji *short* hingga *long-term strategy* berdasarkan kondisi perusahaan yang sekarang. *Chief Operating Officer* (COO) dapat melakukan pengawasan terhadap *day-to-day* operasional sales. *Chief Marketing Officer* (CMO) dapat menentukan rencana *campaign* yang akan dilakukan kedepan dengan melihat performa *existing campaign*

2. Sales Department

Sebagai sarana untuk memonitor performa dan distribusi penjualan serta menganalisis trend yang sedang terjadi pada suatu region, SKU atau kategori tertentu. Selain itu, juga dapat digunakan untuk mengkaji *customer feedback* melalui rating yang diterima.

3. Marketing Department

Sebagai sarana untuk melihat efektifitas dari campaign yang sudah dijalankan serta sarana untuk menganalisis *target market* dari produk yang dipasarkan berdasarkan faktor demografis (tingkat pendidikan, pendapatan, dll).

4. Business Development Department

Sebagai sarana untuk melihat efektifitas dari eksperimen terkait landing page baru yang sedang dikerjakan oleh tim BD.

B. How to solve the problem

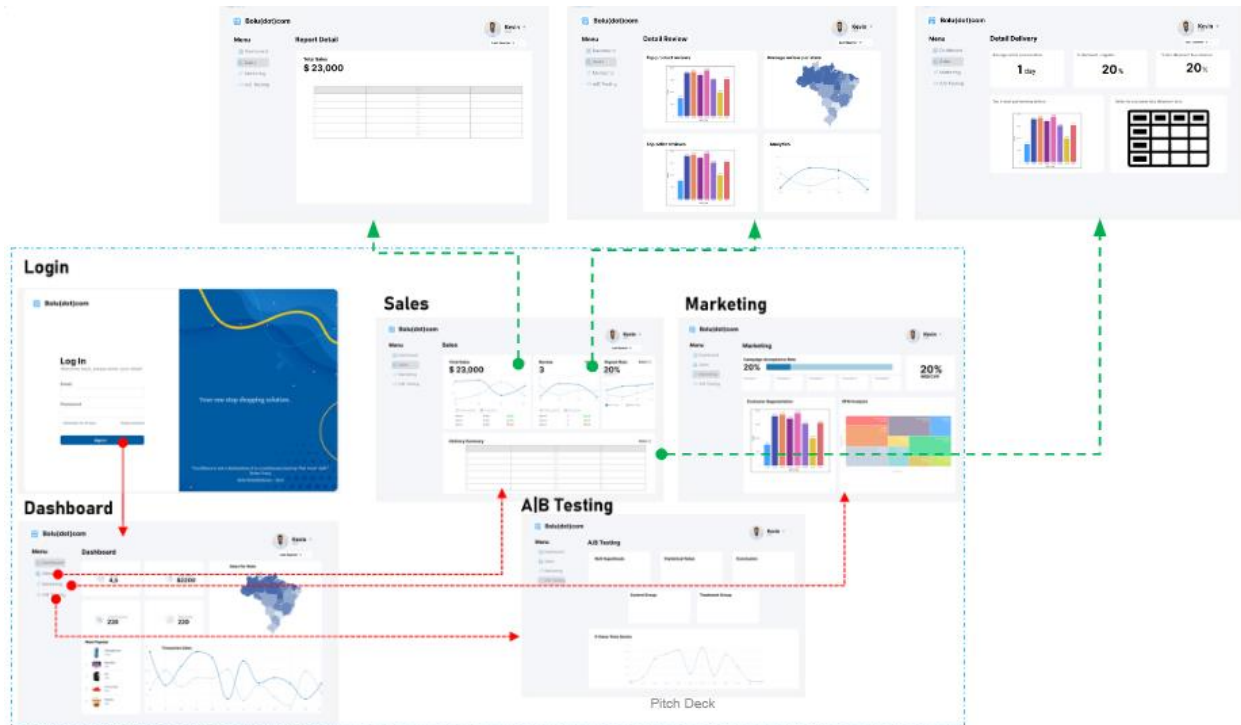
Banyaknya data yang dihasilkan dan adanya urgensi untuk mendapatkan *insight* dari data-data tersebut, maka kami berinisiasi untuk membuat sebuah *tailor-made dashboard* yang spesifik menjawab masing-masing permasalahan *stakeholders*.

Adapun data - data yang akan ditampilkan di *dashboard* antara lain:

- *Sales, customers, orders, top products, performance overview* → untuk memberikan *clear insight* kepada C-Level secara general
- *Operational metrics, review summary, purchase traffic & region per sales* → untuk membantu *sales & operational team* dalam memberikan *detailed explanation* terkait *sales operational, customer satisfaction, ongoing traffic* dan *sales performance*

- *Campaign summary* dan *customer segmentation* dari RFM dan demografi → untuk membantu *marketing team* dalam mempelajari karakteristik, atribut dari masing-masing *customer* dan melakukan *overview* terhadap *campaigns*.
- *A/B testing simple result* → Memberikan konklusi terhadap hasil eksperimen sehingga bisa diambil keputusan.

C. User Flow



Dashboard terdiri 4 bagian utama, yaitu *login page*, *main*, *sales*, *marketing* dan *A/B testing dashboard*. *Login page* adalah fitur untuk *user* memasukkan *email* dan *password* sehingga hanya *stakeholders* tertentu yang bisa masuk ke *dashboard*. Selesai *login*, *user* akan diarahkan ke *main dashboard*. Jika ingin pindah ke *dashboard page* lain, *user* bisa menekan tombol yang terletak di bagian samping.

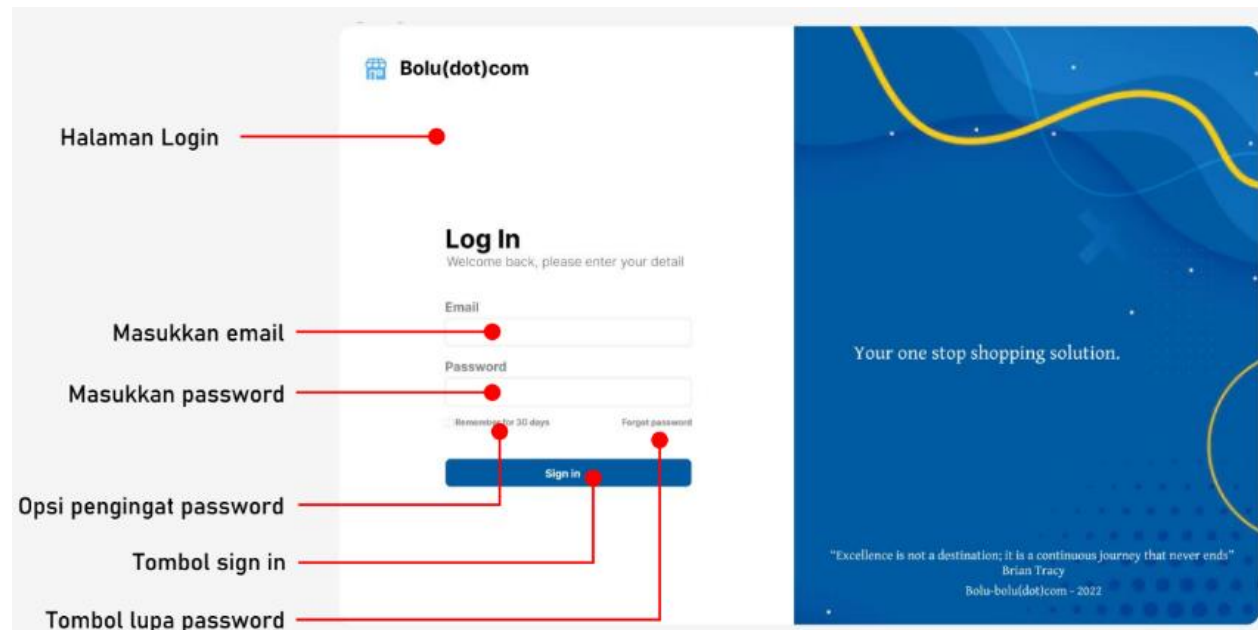
Pada *sales dashboard*, terdapat fitur *button* untuk menampilkan data detail pada segmen tertentu, misalnya *operational metrics*. *Button* ini akan menampilkan *page* baru yang berisikan data pendukung yang menjelaskan *metrics* tersebut, namun ini masih menjadi bagian besar dari *sales dashboard*. Pada masing-masing *detailed dashboard*, terdapat *home button* untuk bisa kembali ke *sales dashboard*.

Marketing dashboard memiliki karakteristik yang sama dengan *sales dashboard*, terdapat juga *detailed dashboard* untuk menunjukkan penjelasan terkait *RFM segmentation*. *A/B testing* tidak memiliki fitur ini. *User* dapat keluar/*logout* dengan menekan tombol *logout*.

D. Penjelasan detail terkait penggunaan dan data pada *dashboard*

Proses penggunaan dan data apa saja yang dapat dilihat dari *dashboard* ini adalah sebagai berikut

a. Login Page



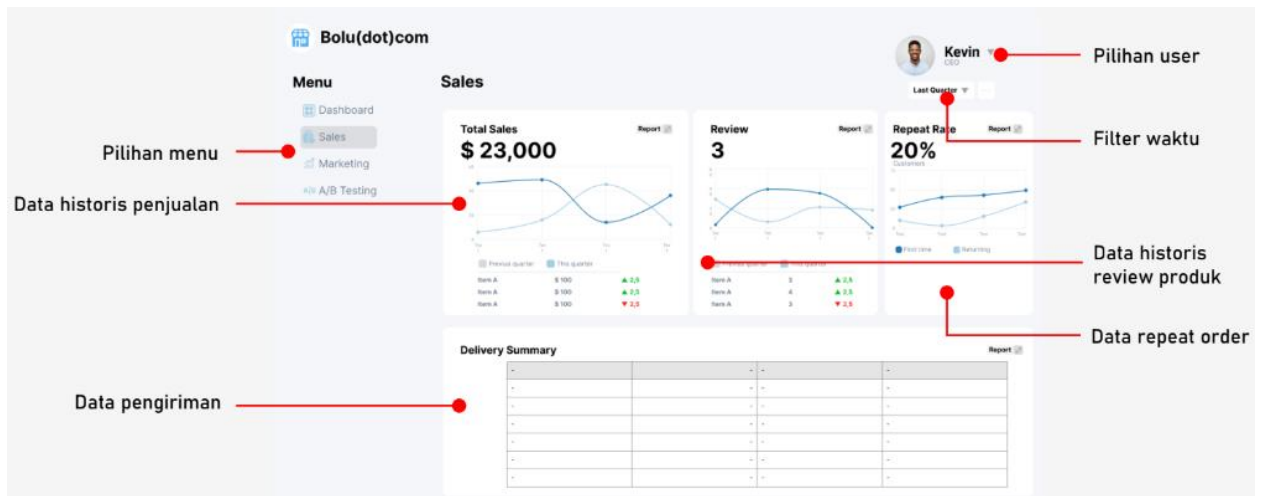
Halaman awal dari *dashboard* yang menuntut *user* untuk melakukan verifikasi dengan cara *login* menggunakan *email* dan *password* yang terdaftar aktif di perusahaan

b. Main Dashboard



Terdapat pilihan menu untuk *user* bisa memilih *directory* dari *dashboard*. Disediakan juga *filter* based on waktu/*region* supaya mengetahui analisis yang lebih detail dari masing-masing grafik/*metrics*.

c. Sales Dashboard



Sales dashboard menampilkan *graph & metrics* yang relevan. Terdapat juga *filtering option* dan juga *detailed data* dengan cara menekan tombol *report*. Saat ditekan, akan memunculkan *detailed dashboard page* sebagai berikut:

Menu

- Dashboard
- Sales
- Marketing
- A/B Testing

Pilihan menu

Detail Delivery

Average order process time

1 day

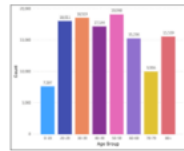
% shipment - logistic

20%

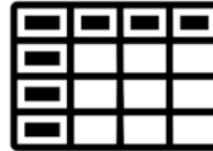
% late shipment to customer

20%

Top / least performing sellers



Seller to customer late shipment rate



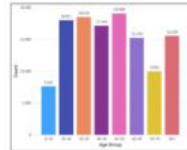
Menu

- Dashboard
- Sales
- Marketing
- A/B Testing

Pilihan menu

Detail Review

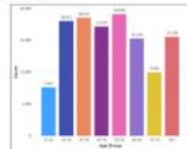
Top product reviews



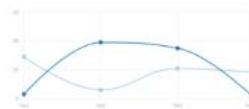
Average review per state



Top seller reviews



Analytics



Menu

- Dashboard
- Sales
- Marketing
- A/B Testing

Pilihan menu

Report Detail

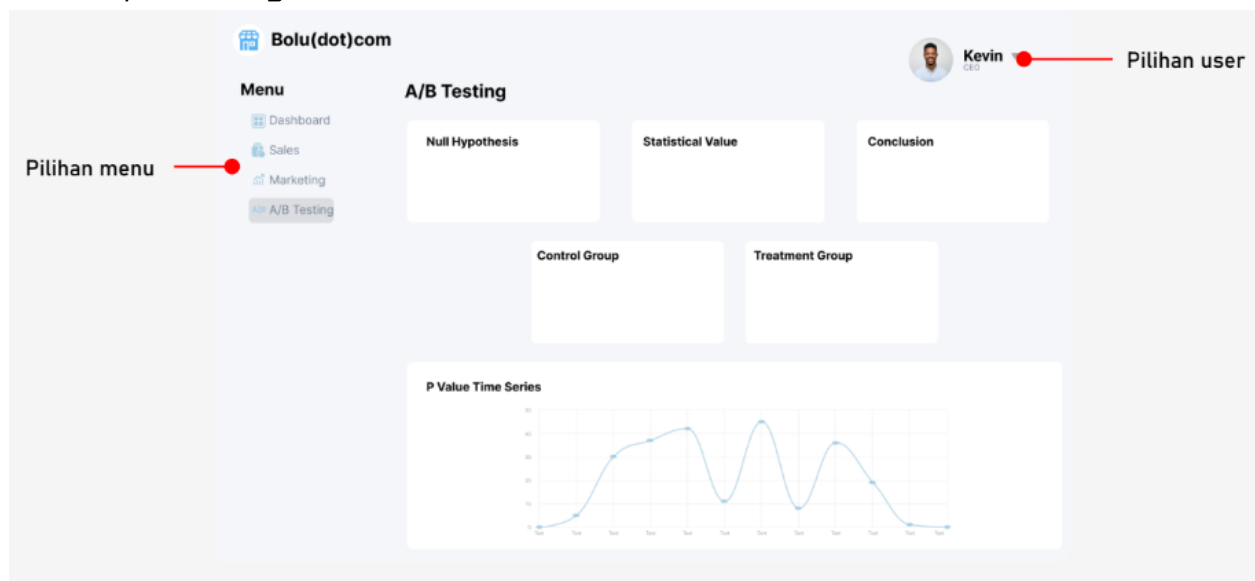
Total Sales
\$ 23,000

Tabel penjualan

d. Marketing Dashboard



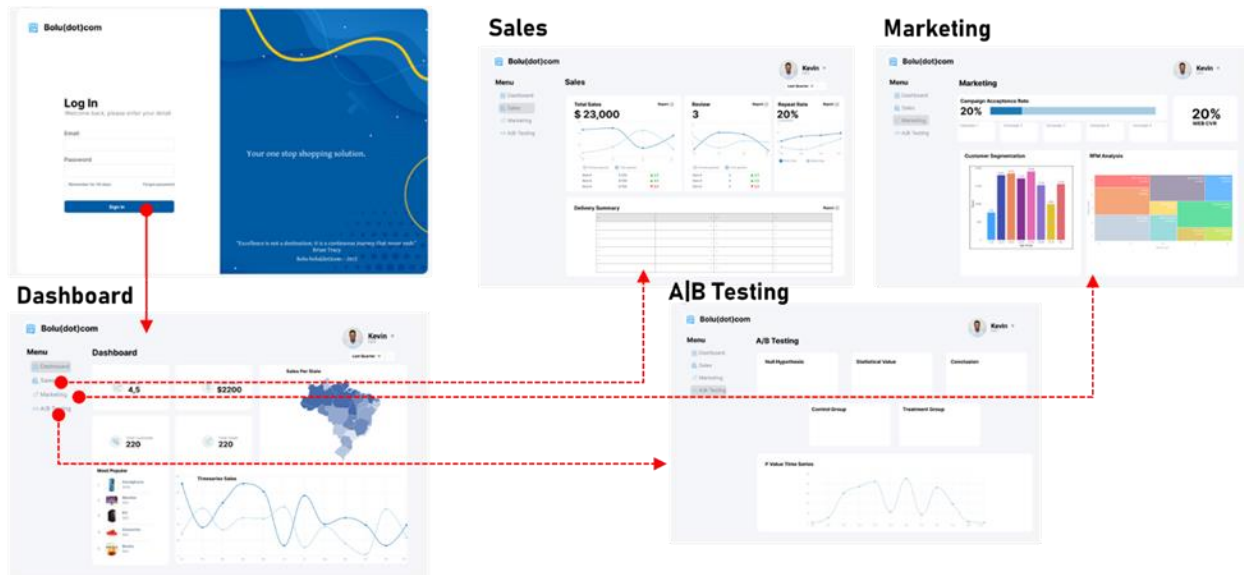
e. A/B Testing



Wireframe

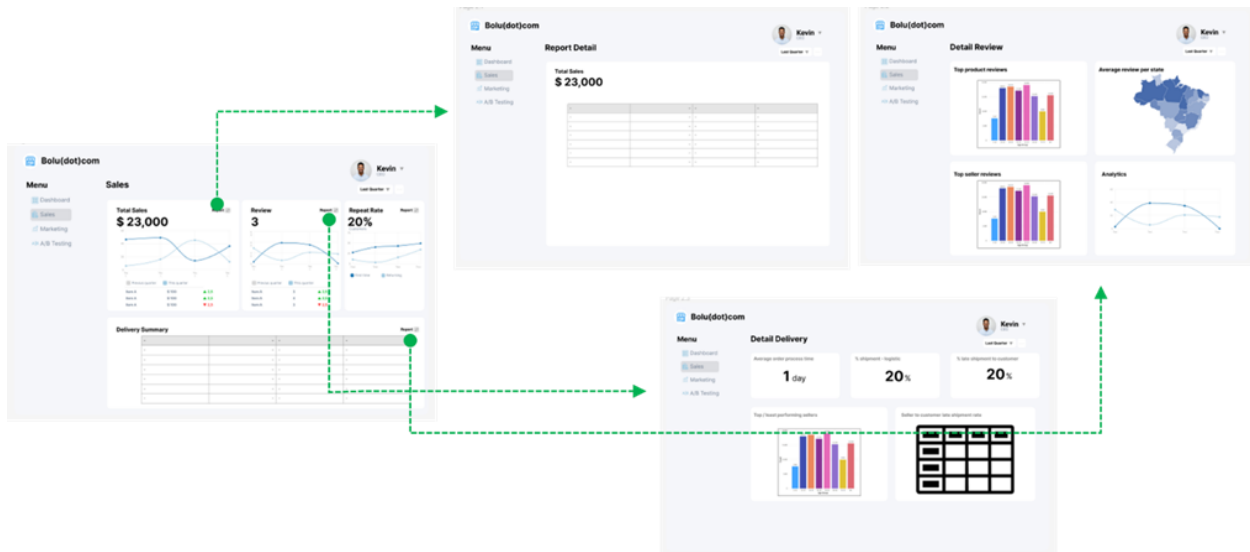
a. Main Dashboard

Login



Gambar diatas memperlihatkan bagaimana user bisa berpindah dari halaman utama (halaman *login*) dari *dashboard* ke bagian yang lebih detail.

b. Detail - Main Dashboard



E. Data Cleaning and Wrangling

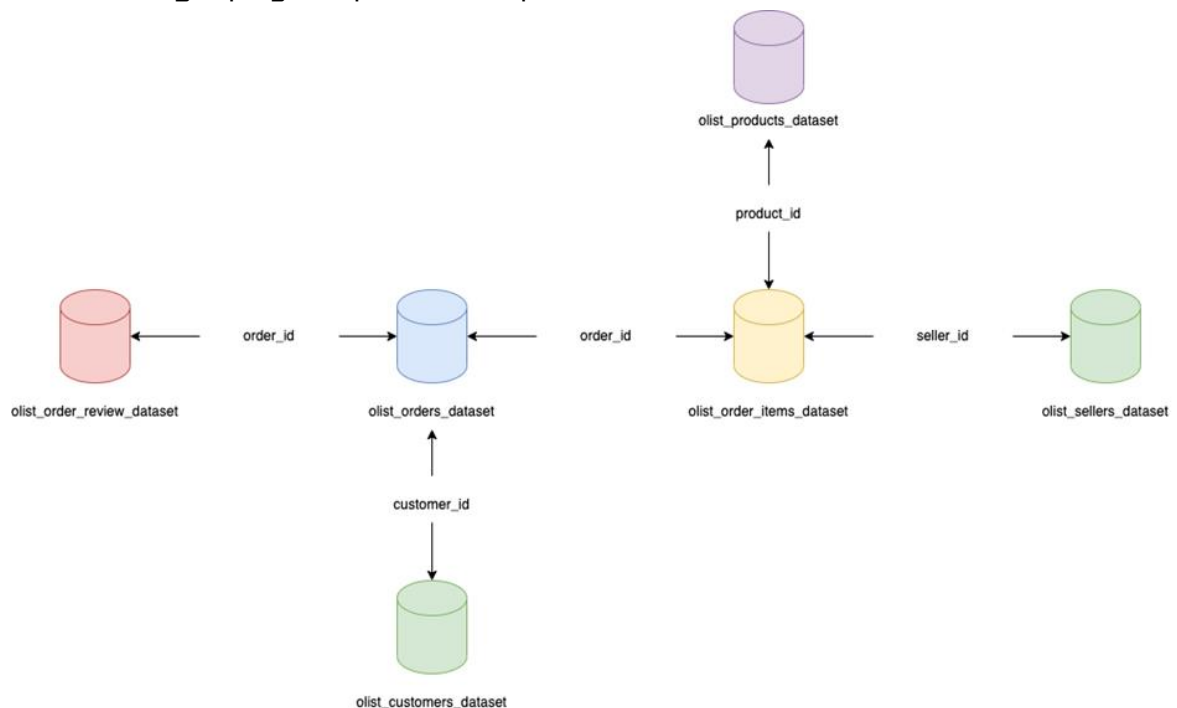
Dalam E-Commerce ini terdapat tiga jenis data utama yang digunakan yaitu:

- Data Sales Performance
- Data Marketing
- Data AB-Testing

Data-data tersebut diekstraksi, ditransformasi dan dirapikan menjadi sebuah data yang 'cleaned' menggunakan bahasa pemrograman python yang akan dijelaskan secara mendetail.

a. Data Sales Performance

Sesuai dengan namanya data sales merupakan data tentang penjualan E-Commerce bolu bolu.com yang memiliki informasi-informasi yang sangat penting seperti data customer, data order, dan, data sellers, data produk, dan data review. Lebih lengkapnya dapat dilihat pada ilustrasi berikut:



Data-data tersebut merupakan data-data terpisah yang harus digabungkan sesuai dengan 'key' yang sudah diberikan sebagai perantara antar-data. Namun sebelum data-data tersebut digabungkan, data akan diperiksa satu per satu untuk eksplorasi secara mendalam. Eksplorasi data dilakukan dengan menggunakan library pandas maupun numpy dalam python.

```
[ ] import pandas as pd
    import numpy as np
```

Berikut adalah contoh eksplorasi data yang dilakukan dalam data sales performance:

- Memeriksa Data Null

```
100*order_review.isna().sum()/len(order_review)
```

```
review_id          0.000000
order_id           0.000000
review_score       0.000000
review_comment_title 88.341530
review_comment_message 58.702532
review_creation_date 0.000000
review_answer_timestamp 0.000000
dtype: float64
```

Order_review merupakan suatu data yang berisikan tentang penilaian seorang customer setelah selesai melakukan order. Gambar 3 menunjukkan berapa persen jumlah data kosong (data NA) yang terdapat pada setiap kolom data order_review. Pada kolom review_comment_title terdapat sekitar 88% data yang kosong dan sekitar 58% untuk kolom review_comment_message. Dalam hal ini kedua data tersebut sebaiknya di drop karena selain memakan waktu yang banyak untuk melakukan proses translate, review-review tersebut juga kurang mewakili seluruh transaksi yang ada.

```
order_review.drop(columns=['review_comment_title', 'review_comment_message'], axis=1, inplace=True)
```

- Translate data nama produk kategori

Dalam proses ini dibutuhkan suatu fungsi yang didapat dalam modul translators.

b. Data Marketing

Description

ID	User ID
Year Birth	User Birth Year
Education	User Education
Marital Status	User Marital Status
Income	User Income (\$)
Dt_Custom	Date of customer's enrollment with the company
Recency	Number of days since customer's last purchase
MntWines	Amount spent on wine in the last 2 years
MntFruits	Amount spent on fruits in the last 2 years
MntMeatProducts	Amount spent on meat in the last 2 years
MntFishProducts	Amount spent on fish in the last 2 years
NumWebPurchases	Number of purchases made through the company's web site
NumWebVisitsMonth	Number of visits to company's web site in the last month
AcceptedCmp3	1 if customer accepted the offer in the 3rd campaign, 0 otherwise
AcceptedCmp4	1 if customer accepted the offer in the 4th campaign, 0 otherwise
AcceptedCmp5	1 if customer accepted the offer in the 5th campaign, 0 otherwise
AcceptedCmp1	1 if customer accepted the offer in the 2nd campaign, 0 otherwise
AcceptedCmp2	1 if customer accepted the offer in the 1st campaign, 0 otherwise

Dalam sebuah E-Commerce, marketing merupakan salah satu department yang sangat menentukan bagaimana performance perusahaan secara keseluruhan. Dalam mengukur marketing performance terdapat sebuah 'metric' yang dapat menjadi indikator yaitu RFM analysis.

Untuk melakukan RFM analysis, dibutuhkan beberapa data yaitu:

- Recency (R)
Untuk mengukur tingkat 'kapan terakhir' customer melakukan pembayaran. Semakin tinggi Recency berarti 'sudah lama' customer melakukan pembayaran terakhir.
- Frequency (F)
Untuk mengukur seberapa sering customer melakukan pembayaran. Pada deskripsi data di atas, NumWebPurchases merupakan Frequency dalam RFM analysis.
- Monetary (M)
Untuk mengukur seberapa banyak uang yang telah digunakan dalam pembayaran. Pada deskripsi data marketing di atas nilai monetary didapatkan dengan menjumlahkan MntWines, MntFruits, MntMeatProducts, MntFishProducts.

Setelah mendapatkan nilai RFM kemudian akan dicari rating dari setiap nilai dengan wrangling data.

```
## membuat rating recency dimulai dari bins untuk rating 0-14 < 14-30 < 30-45 < 45-75 < 75-100
marketing['Recency_Rating'] = pd.cut(x=marketing['Recency'],
    bins=[0, 14, 30, 45, 75, 99],
    labels=['5', '4', '3', '2', '1'], ## Label harus menerima tipe data str
    include_lowest=True, ## supaya nilai terendah (0) ikut masuk bin
)

## membuat rating monetary dimulai dari bins untuk rating (<100) < 100-499 < 500-999 < 1000-2000 < (>2000)
marketing['Monetary_Rating'] = pd.cut(x=marketing['Monetary'],
    bins=[0, 100, 500, 1000, 2001, np.inf],
    labels=['1', '2', '3', '4', '5'], ## Label harus menerima tipe data str
    right = False)

## membuat rating frequency dimulai dari bins untuk rating (<5) < 5-9 < 10-14 < 15-20 < (>21)
marketing['Frequency_Rating'] = pd.cut(x=marketing['NumWebPurchases'],
    bins=[0, 5, 10, 15, 21, np.inf],
    labels=['1', '2', '3', '4', '5'], ## Label harus menerima tipe data str
    right = False)
```

Masing-masing dari R,F, dan M akan dibagi-bagi menjadi 5 kelompok yang detailnya dapat dilihat pada gambar (tulisan berwarna hijau). Untuk rating pada Recency, semakin 'baru' (0-14) customer melakukan pembayaran maka customer akan mendapatkan recency rating tinggi (5). Untuk rating pada Monetary, semakin banyak customer 'menghabiskan' (>2000) uangnya maka akan

mendapatkan monetary rating yang tinggi (5). Untuk rating pada Frequency, semakin sering customer melakukan transaksi (>21) maka akan mendapatkan Frequency rating yang tinggi (5). RFM rating merupakan rata-rata rating dari R,F, dan M yang dibulatkan.

```
[ ] marketing['RFM_Rating'] = marketing[['Recency_Rating', 'Frequency_Rating', 'Monetary_Rating']].astype(int).mean(axis=1).apply(round)
marketing
```

Recency_Rating	Monetary_Rating	Frequency_Rating	RFM_Rating
5	3	1	3
5	3	2	3
5	2	1	3
5	1	1	2
5	1	1	2
...
1	3	2	2
1	1	1	1
1	2	2	2
1	4	2	2
1	3	2	2

Selanjutnya untuk mendapatkan customer label digunakan syntax berikut:

```
] ## Customer Labelling menggunakan regex
segmentasi_map = {
    r'[1-2][1-2]': 'hibernating',
    r'[1-2][3-4]': 'at_Risk',
    r'[1-2]5': 'cannot_loose_them',
    r'3[1-2]': 'about_to_sleep',
    r'33': 'need_attention',
    r'[3-4][4-5]': 'loyal_customers',
    r'41': 'promising',
    r'51': 'new_customers',
    r'[4-5][2-3]': 'potential_loyalists',
    r'5[4-5]': 'champions'
}

marketing['Customer_Label'] =(marketing['Recency_Rating'].astype(str) + marketing['Frequency_Rating'].astype(str)).replace(segmentasi_map, regex=True)
marketing
```

c. Data AB-Testing

Data AB Test merupakan data hasil eksperimen terhadap user untuk menguji apakah new landing page memiliki conversion rate yang lebih tinggi secara signifikan dibandingkan dengan old landing page.

```
[ ] df = pd.read_csv('ab.csv')
    df.sort_values(by='timestamp')
```

	user_id	timestamp	group	landing_page	converted
131228	922696	2017-01-02 13:42:05.378582	treatment	new_page	0
184884	781507	2017-01-02 13:42:15.234051	control	old_page	0
83878	737319	2017-01-02 13:42:21.786186	control	old_page	0
102717	818377	2017-01-02 13:42:26.640581	treatment	new_page	0
158789	725857	2017-01-02 13:42:27.851110	treatment	new_page	0
...
153305	851645	2017-01-24 13:41:18.869978	treatment	old_page	0
47535	808330	2017-01-24 13:41:19.152664	control	old_page	0
157787	843121	2017-01-24 13:41:44.097174	treatment	new_page	0
179072	836373	2017-01-24 13:41:52.604673	control	old_page	0
193652	920411	2017-01-24 13:41:54.460509	control	old_page	0

294478 rows × 5 columns

Untuk mendapatkan hasil data AB Test diperlukan beberapa 'bahan-bahan' seperti:

- Jumlah Control

Control merupakan user telah diarahkan (visit) ke old-page. Dalam hal ini akan didapatkan jumlah visit dan jumlah conversion, dimana conversion merupakan banyaknya user yang 'akhirnya' melakukan purchase

- Jumlah Treatment

Treatment merupakan user yang telah diarahkan (visit) ke new-page. Seperti halnya dengan control, pada treatment juga akan didapatkan jumlah visit dan jumlah conversion.

Kolom landing_page sejatinya sama dengan kolom group oleh karena itu kolom landing_page tidak akan digunakan dalam proses AB-Testing karena kolom group merupakan data yang bahasanya sesuai dengan terminologi AB-Test.

Dengan wrangling data maka akan didapatkan banyak jumlah control, dan treatment per hari dengan masing-masing terdiri atas visit dan conversion. Berikut merupakan contoh 2 hari percobaan AB-Test.

```
valuenya = ab_test.groupby(['date', 'group']).agg(['count', 'sum'])  
valuenya
```

		converted	
		count	sum
date	group		
2017-01-02	control	2894	362
	treatment	2889	348
2017-01-03	control	6684	760
	treatment	6710	762

Untuk mendapatkan jumlah visit pada Gambar diatas proses agregasi menggunakan Count terhadap kolom Converted untuk sedangkan untuk mendapatkan jumlah conversion digunakan fungsi agregasi sum, untuk menjumlahkan nilai 1 yang terdapat pada kolom Converted. Setelah mendapatkan data perhari group, treatment dengan masing-masing terdiri atas visit (count) dan conversion maka data akan ditransformasi hingga mendapatkan jumlah control dan treatment yang telah terakumulasi dari hari ke hari (date by date) sampai pada akhir AB-Testing.


```
[ ] ab_test_ready = ab_test.set_index('date_ab_test').cumsum() #cumulative sum
ab_test_ready
```

	control_visit	treatment_visit	control_converted	treatment_converted
date_ab_test				
2017-01-02	2894	2889	362	348
2017-01-03	9578	9599	1122	1110
2017-01-04	16233	16228	1932	1883
2017-01-05	22771	22814	2733	2641
2017-01-06	29455	29658	3511	3489
2017-01-07	36133	36361	4316	4265
2017-01-08	42902	43156	5121	5086
2017-01-09	49627	49870	5927	5880
2017-01-10	56372	56648	6690	6738
2017-01-11	63159	63414	7498	7523
2017-01-12	69774	70121	8307	8343
2017-01-13	76423	76710	9086	9077
2017-01-14	83058	83404	9928	9876
2017-01-15	89867	90044	10751	10633
2017-01-16	96550	96688	11567	11428
2017-01-17	103241	103319	12385	12272
2017-01-18	109819	110026	13205	13107
2017-01-19	116484	116654	14000	13887
2017-01-20	123110	123421	14763	14684
2017-01-21	129933	130073	15622	15456
2017-01-22	136608	136821	16418	16251
2017-01-23	143391	143549	17272	17065
2017-01-24	147202	147276	17723	17514

Pada akhirnya data tersebut akan digunakan untuk mencari nilai p dan nilai power dimana kedua nilai tersebut menjadi hasil untuk menarik kesimpulan apakah new_landing_page dapat meningkatkan conversion rate secara signifikan dibandingkan dengan old_page.

Dalam mendapatkan nilai p dan nilai power akan dimanfaatkan sebuah library statistik kemudian akan dibuat suatu fungsi untuk mendapatkan nilai p

dan nilai power yang dapat di apply dalam data AB-Test yang sudah terakumulasi.

```
from statsmodels.stats.proportion import power_proportions_2indep, test_proportions_2indep
from statsmodels.stats.weightstats import ttest_ind
from statsmodels.stats.power import tt_ind_solve_power
```

```
[ ] def p_value(ab_test_ready):
    result = test_proportions_2indep(count1=ab_test_ready["treatment_converted"],
                                     nobs1=ab_test_ready["treatment_visit"],
                                     count2=ab_test_ready["control_converted"],
                                     nobs2=ab_test_ready["control_visit"],
                                     alternative="two-sided")
    return result[1]
```

```
[ ] def power(ab_test_ready):
    test_split = ab_test_ready['treatment_visit'] / (ab_test_ready['treatment_visit'] + ab_test_ready['control_visit'])
    relative_effect = (ab_test_ready['treatment_converted_rate'] - ab_test_ready['control_converted_rate']) / ab_test_ready['control_converted_rate']

    power_result = power_proportions_2indep(diff=ab_test_ready['control_converted_rate'] * relative_effect,
                                             prop2=ab_test_ready['control_converted_rate'],
                                             nobs1=ab_test_ready['treatment_visit'],
                                             ratio=(1-test_split)/test_split,
                                             alpha=0.05,
                                             alternative='two-sided',
                                             return_results=False)

    return power_result
```

```
[ ] ab_test_ready['p_value'] = ab_test_ready.apply(p_value, axis=1)
    ab_test_ready['power'] = ab_test_ready.apply(power, axis=1)
```

Dalam parameter fungsi power terdapat converted_rate dimana nilai ini merupakan nilai yang didapatkan dari (jumlah conversion / jumlah visit).

	control_visit	treatment_visit	control_converted	treatment_converted	control_converted_rate	treatment_converted_rate	p_value	power
date_ab_test								
2017-01-02	2894	2889	362	348	0.125086	0.120457	0.592186	0.081031
2017-01-03	9578	9599	1122	1110	0.117143	0.115637	0.745064	0.061902
2017-01-04	16233	16228	1932	1883	0.119017	0.116034	0.404180	0.128529
2017-01-05	22771	22814	2733	2641	0.120021	0.115762	0.158646	0.274735
2017-01-06	29455	29658	3511	3489	0.119199	0.117641	0.557845	0.089255
2017-01-07	36133	36361	4316	4265	0.119448	0.117296	0.369942	0.142723
2017-01-08	42902	43156	5121	5086	0.119365	0.117852	0.492331	0.104243
2017-01-09	49627	49870	5927	5880	0.119431	0.117907	0.457241	0.113778
2017-01-10	56372	56648	6690	6738	0.118676	0.118945	0.888844	0.052258
2017-01-11	63159	63414	7498	7523	0.118716	0.118633	0.963503	0.050240
2017-01-12	69774	70121	8307	8343	0.119056	0.11898	0.965077	0.050220
2017-01-13	76423	76710	9086	9077	0.118891	0.118329	0.733720	0.063257
2017-01-14	83058	83404	9928	9876	0.119531	0.118412	0.480630	0.107649
2017-01-15	89867	90044	10751	10633	0.119632	0.118087	0.311124	0.170140
2017-01-16	96550	96688	11567	11428	0.119803	0.118195	0.274866	0.189992
2017-01-17	103241	103319	12385	12272	0.119962	0.118778	0.406532	0.130265
2017-01-18	109819	110026	13205	13107	0.120243	0.119126	0.419850	0.125838
2017-01-19	116484	116654	14000	13887	0.120188	0.119044	0.394807	0.134566
2017-01-20	123110	123421	14763	14684	0.119917	0.118975	0.470737	0.110493
2017-01-21	129933	130073	15622	15456	0.120231	0.118826	0.269312	0.193849
2017-01-22	136608	136821	16418	16251	0.120183	0.118776	0.256515	0.201998
2017-01-23	143391	143549	17272	17065	0.120454	0.118879	0.193835	0.249700
2017-01-24	147202	147276	17723	17514	0.120399	0.11892	0.216126	0.230949

```
] ab_test_ready.to_excel('ab_test_result.xlsx')
```

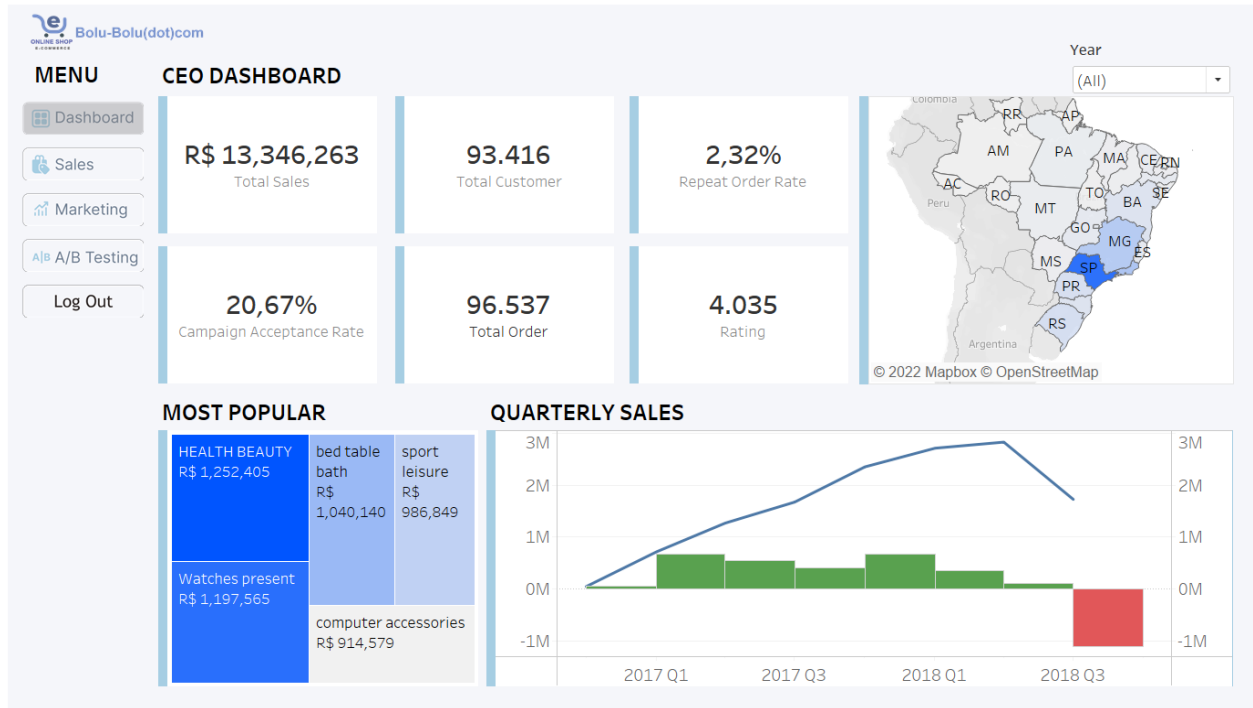
Pada akhirnya akan didapat p_value dan power yang merupakan hasil dari suatu AB-Testing. Data ini akan di-export dalam bentuk excel untuk digunakan dalam visualisai Tableau.

- Penggabungan Data

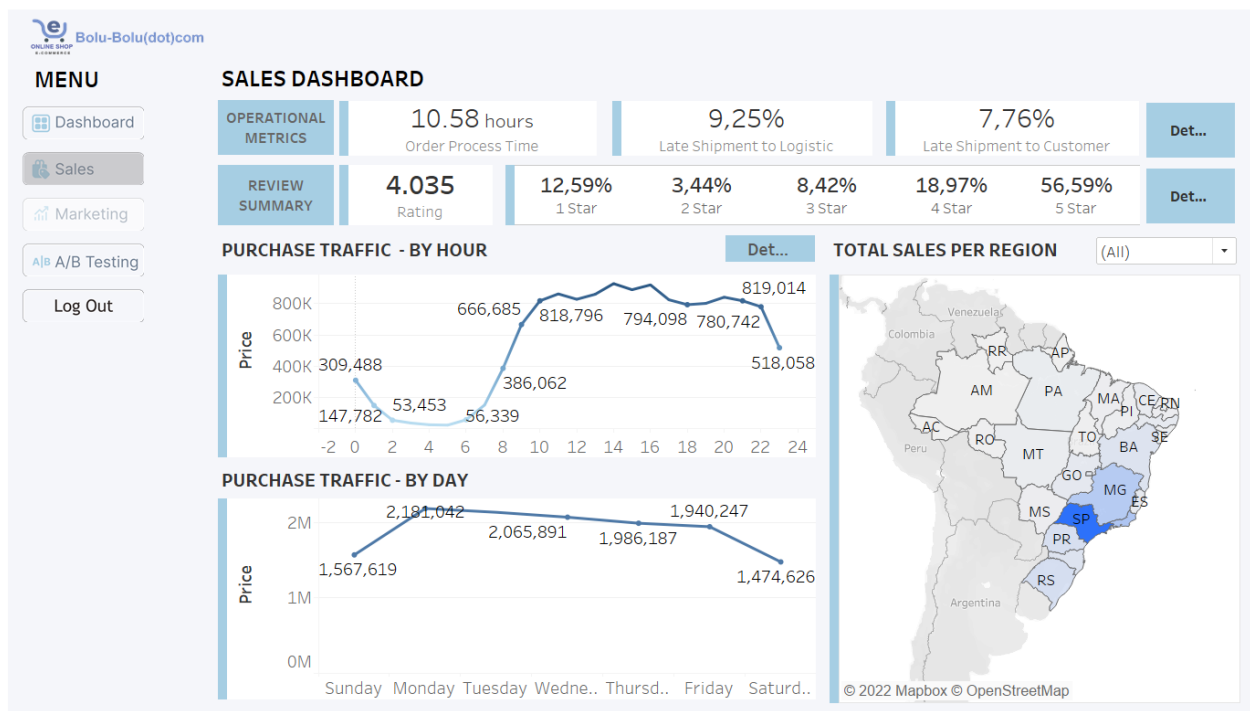
Setelah melakukan wrangling untuk setiap data. Ketiga data tersebut akan di-merged dengan fungsi pandas yang sama tetapi dengan menggunakan metode **outer join** dengan key yang digunakan adalah index yang dimiliki oleh setiap data.

F. Analisis

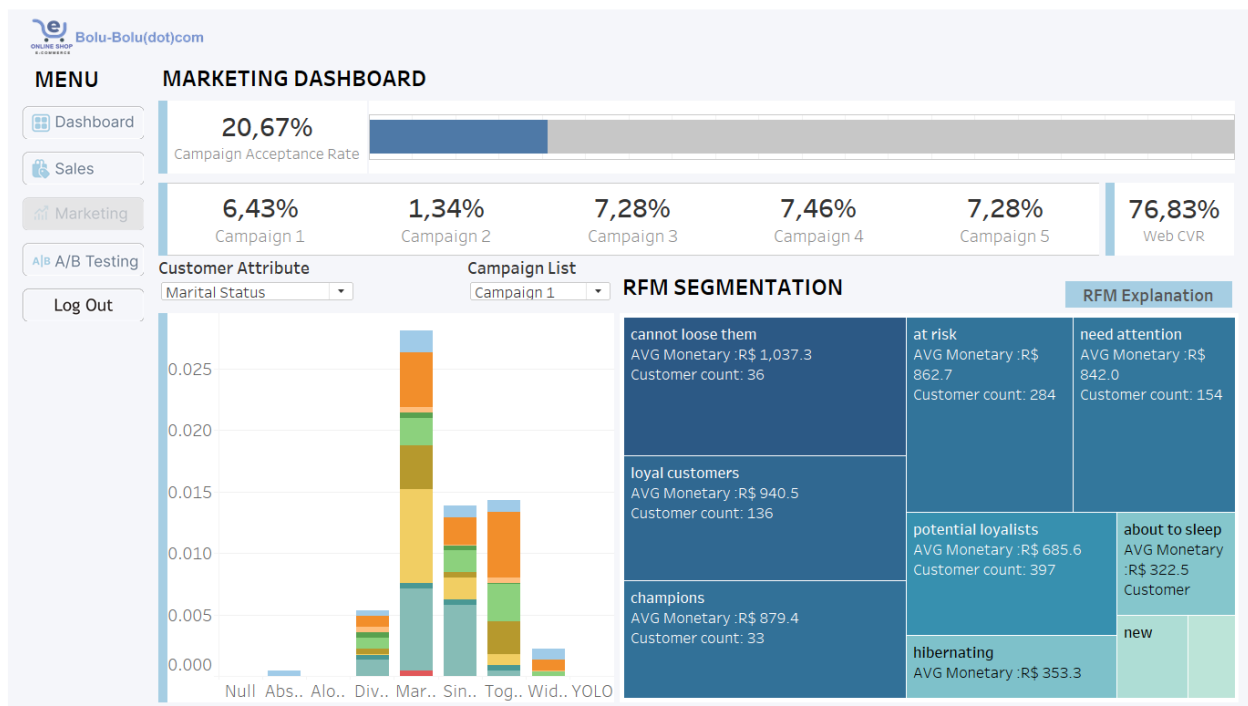
Dari *dashboard* pertama, yaitu *CEO* dashboard dapat dilihat beberapa insight yang dapat membantu CEO untuk melihat kondisi perusahaan secara menyeluruh.



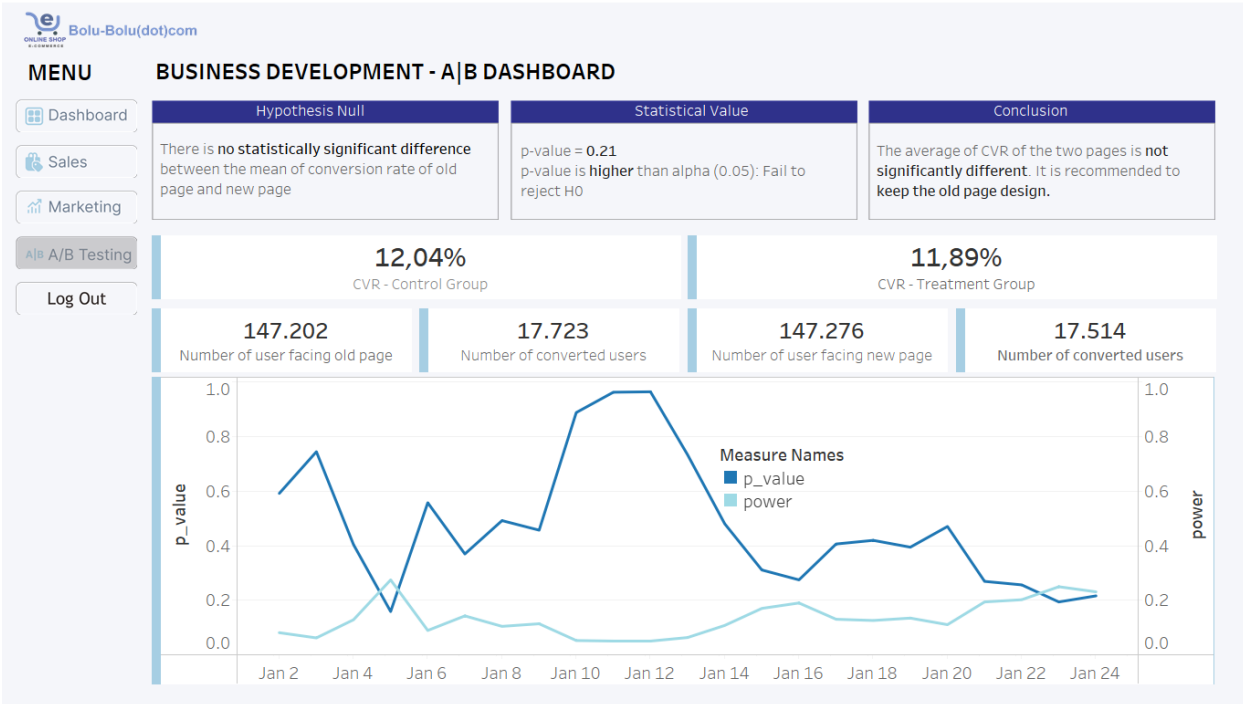
Dengan bantuan *dashboard* ini, CEO dapat mengetahui jumlah penjualan, customer, *campaign acceptance rate*, hingga *repeat order rate* dalam satu tempat. Selain itu juga CEO juga dapat melihat data historis berdasarkan *quarter*.



Dari sales *dashboard* baik CEO maupun departemen terkait dapat melihat kondisi penjualan dengan lebih detail. *Dashboard* ini memperlihatkan bagaimana *traffic* pembelian berdasarkan jam dan hari pembelian di tiap - tiap kota di Brazil. Selain itu, dari dashboard ini juga dapat dilihat seberapa besar persentase keterlambatan pengiriman sampai ke pembeli untuk materi *improvement* dari sisi tim sales.



Marketing *dashboard* memberikan informasi bahwa sampai saat ini *campaign acceptance rate* hanya sekitar 20,67% sehingga dapat menjadi masukan untuk tim marketing dalam proses pembuatan *campaign* berikutnya.



Dari *dashboard* Bisdev ini, tim business development bisa melihat bahwa experiment terkait rencana penggantian landing page baru, berdasarkan hasil A/B testing, dapat disimpulkan bahwa rata - rata *conversion rate* dari dua halaman / *landing page* yang di eksperimenkan tidak berbeda secara signifikan. Sehingga disimpulkan untuk tetap menggunakan desain yang lama.

G. Link Github

<https://github.com/kevreinhard>

<https://github.com/hendrayogawiguna>

<https://github.com/billyaja/E-Commerce>