

IMPLEMENTASI *TWILIO SEGMENT* DAN *TABLEAU* UNTUK MENINGKATKAN PROSES ANALISIS DATA PELANGGAN DI *MODERNVET*

Ridwan Zulkifli¹, Maulana Nurkarim²
ridwan.zulkifli12@gmail.com
Universitas Ma'soem

Abstract

This research focuses on the implementation of Twilio Segment and data visualization using Tableau to enrich customer data analysis at Modernvet, a platform used by pet owners to access various services related to their pet's health care. The application encompasses features such as scheduling veterinarian appointments, managing appointments, viewing health test results, and bill payments. By integrating Twilio Segment, customer data from various sources can be gathered efficiently. The results of data analysis translated into visualizations with Tableau provide deeper insights and support better decision-making. The outcomes of this implementation have the potential to enhance services and the customer experience at Modernvet.

Keywords: *Twilio Segment; Tableau; Customer Data Analysis; Data Visualization; Modernvet*

Abstrak

Penelitian ini fokus pada implementasi *Twilio Segment* dan visualisasi data menggunakan *Tableau* untuk meningkatkan analisis data pelanggan di *Modernvet*, sebuah aplikasi *mobile* yang digunakan oleh pemilik hewan peliharaan untuk mengakses berbagai layanan terkait kesehatan hewan peliharaan mereka. Aplikasi ini mencakup fitur seperti jadwal dokter hewan, pengelolaan janji temu, melihat hasil tes kesehatan, dan pembayaran tagihan. Dengan mengintegrasikan *Twilio Segment*, data pelanggan dari berbagai sumber dapat dihimpun secara efisien. Hasil analisis data yang diterjemahkan ke dalam visualisasi dengan *Tableau* memberikan wawasan yang lebih dalam dan mendukung pengambilan keputusan yang lebih baik. Hasil implementasi ini memiliki potensi untuk meningkatkan layanan dan pengalaman pelanggan di *Modernvet*.

Keywords: *Twilio Segment; Tableau; Analisis Data Pelanggan; Visualisasi Data; Modernvet*

Pendahuluan

Penelitian ini bertujuan untuk mengatasi tantangan yang dihadapi oleh *Modernvet*, sebuah aplikasi *mobile* layanan kesehatan hewan peliharaan, dalam mengelola dan menganalisis data pelanggan. Masalah yang menjadi fokus penelitian ini adalah kurangnya efisiensi dalam mengintegrasikan data pelanggan dari berbagai sumber yang beragam. Harapan perusahaan adalah untuk memberikan

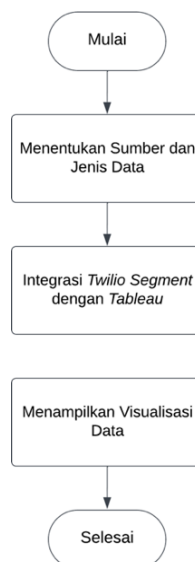
layanan yang lebih baik dan lebih terpersonalisasi kepada pemilik hewan peliharaan melalui analisis data yang baik.

Kesenjangan ini menjadi penyebab utama masalah yang akan diteliti, yaitu bagaimana meningkatkan efisiensi pengelolaan data pelanggan di *Modernvet*. Untuk memecahkan masalah ini, penelitian ini mengusulkan implementasi *Twilio Segment* sebagai alat untuk mengumpulkan dan mengintegrasikan data dari berbagai sumber. Selanjutnya, data tersebut akan dianalisis dan divisualisasikan menggunakan perangkat *Tableau*. Dengan solusi ini, diharapkan dapat meningkatkan pemahaman terhadap perilaku pelanggan, meningkatkan layanan, dan mendukung pengambilan keputusan yang lebih cerdas berdasarkan data pelanggan yang lebih kaya.

Tujuan utama adalah untuk mengimplementasikan *Twilio Segment* sebagai alat untuk mengumpulkan dan mengintegrasikan data pelanggan dari berbagai sumber di *Modernvet* dan Menerapkan *Tableau* untuk memvisualisasikan data pelanggan dengan tujuan untuk memberikan wawasan yang lebih mendalam kepada *Modernvet*.

Metode Penelitian

Metode penelitian adalah tahapan - tahapan dalam penelitian yang berfungsi untuk memecahkan suatu permasalahan agar pelaksanaan penelitian sesuai dengan tujuan. Metode penelitian yang akan dilakukan pada penelitian ini dapat dilihat pada Gambar 1 berikut:



Gambar 1 Tahapan Metode Penelitian

a. Menentukan Sumber Dan Jenis Data

Pada tahapan pertama dilakukan penentuan sumber dan jenis data. Data pelanggan dalam Aplikasi *Modernvet* diakses melalui tiga jenis sumber utama:

1. Sumber Data Dari In-App Events

Data ini adalah data yang berasal dari acara dalam aplikasi yang dicatat setiap

kali pengguna melakukan tindakan, seperti membuat janji temu atau melakukan pembayaran. Data ini dikumpulkan secara real-time melalui metode Segment Event Tracking.

2. Sumber Data Dari Vetlink

Vetlink adalah platform terintegrasi yang digunakan dalam Aplikasi Modernvet. Data pelanggan, termasuk data tagihan, data janji temu, data staf (dokter hewan), data produk, data pelanggan (pemilik hewan), dan data pasien (hewan), tersimpan di dalam sistem Vetlink. Untuk mengakses data dari Vetlink, penulis menggunakan API yang disediakan oleh Vetlink.

3. Sumber Data Dari Database

Beberapa data, khususnya data pembayaran yang berasal dari vendor payment gateway dari Indonesia yaitu Xendit, disimpan dalam database internal. Penulis mengakses data ini langsung dari database menggunakan query SQL.

Analisis data pelanggan melibatkan berbagai jenis data, termasuk:

1. Data Tagihan

Data ini mencakup informasi mengenai tagihan yang telah dibuat kepada pemilik hewan, termasuk jumlah, tanggal, dan status pembayaran.

2. Data Janji Temu

Data ini mencakup jadwal janji temu antara pemilik hewan dan dokter hewan, termasuk tanggal, waktu, dan keterangan tambahan.

3. Data Staf (Dokter Hewan)

Data ini mencakup informasi mengenai staf medis yang terlibat dalam perawatan hewan, termasuk nama, spesialisasi, dan jadwal kerja.

4. Data Produk

Data ini mencakup inventaris produk dan obat-obatan yang tersedia di klinik, termasuk harga, stok, dan deskripsi.

5. Data Pelanggan (Pemilik Hewan Peliharaan)

Data ini adalah data tentang pemilik hewan, termasuk nama, alamat, nomor telepon, dan informasi kontak lainnya.

6. Data Pasien (Hewan Peliharaan)

Data ini mencakup informasi tentang hewan peliharaan, termasuk nama, jenis hewan, riwayat kesehatan, dan catatan medis.

7. Data Pembayaran

Data pembayaran yang berasal dari vendor *Xendit* digunakan untuk melacak semua transaksi pembayaran yang terkait dengan layanan dalam Aplikasi *Modernvet*.

8. Data *In-App Events*

Data *In-App Events* adalah data acara aplikasi yang memungkinkan perusahaan untuk mendapatkan wawasan yang lebih baik tentang perilaku pelanggan dan tren yang mungkin terjadi.

b. Integrasi Twilio Segment dengan Tableau

Tahap berikutnya adalah Integrasi Twilio Segment dengan Tableau. Pada tahap ini memerlukan beberapa langkah yang perlu dilakukan:

1. Konfigurasi Twilio Segment

Langkah ini mencakup pengaturan sumber data untuk mengumpulkan data dari berbagai sumber yang ada dan juga menentukan destinasi dari hasil

pengumpulan data.

2. Integrasi Twilio Segment dengan Sumber Data

Data dari sumber - sumber yang berbeda akan diintegrasikan dengan Twilio Segment menggunakan alat bantu atau metode yang telah dipilih. Data akan diproses dan disatukan dalam Twilio Segment untuk kemudian dikirim ke destinasi data yang telah di tentukan pada langkah sebelumnya.

3. Menghubungkan Destinasi Data dengan Tableau

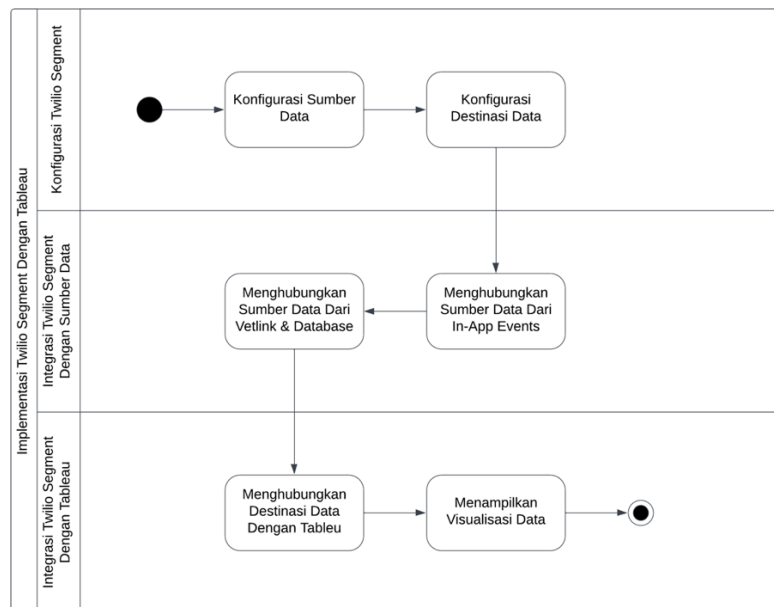
Langkah selanjutnya menghubungkan destinasi data dengan perangkat visualisasi Tableau. Ini akan memungkinkan penggunaan Tableau untuk analisis data pelanggan yang lebih mendalam dan pembuatan visualisasi yang informatif.

c. Menampilkan Visualisasi Data

Tahap akhir akan dijelaskan salah satu contoh bagaimana hasil analisis data pelanggan dapat diwujudkan dalam bentuk visualisasi yang informatif dan mudah dipahami.

Hasil dan Pembahasan

Implementasi *system* yang telah dirancang pada bab sebelumnya. Implementasi *system* ini meliputi implementasi program. Dapat dilihat dari model yang digambarkan melalui *Activity Diagram* di bawah ini:



Gambar 2 Activity Diagram

1. Konfigurasi Twilio Segment

Twilio Segment memerlukan konfigurasi untuk memastikan pengumpulan dan pengelolaan data pelanggan berjalan efisien. Pada tahap ini, fokus akan diberikan pada dua aspek utama, yaitu konfigurasi sumber data dan destinasi data.

a. Konfigurasi Sumber Data

Langkah pertama dalam konfigurasi *Twilio Segment* adalah menetapkan jenis sumber data. Dua jenis sumber data yang terlibat dalam proses ini adalah:

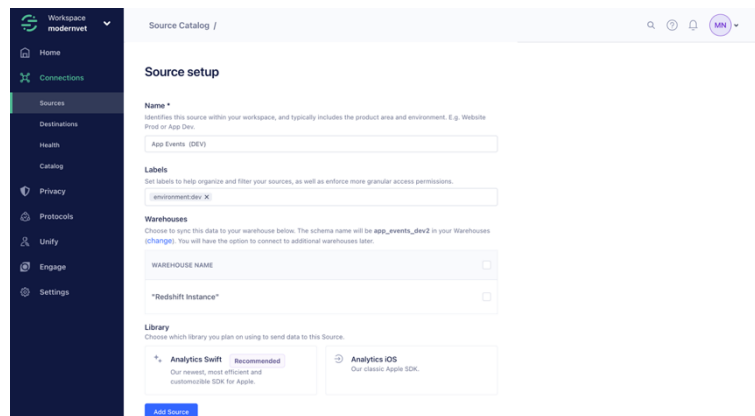
1. *App Event*

Konfigurasi ini memungkinkan *Twilio Segment* dapat mengumpulkan data yang berkaitan dengan interaksi pelanggan di aplikasi *Modernvet*.

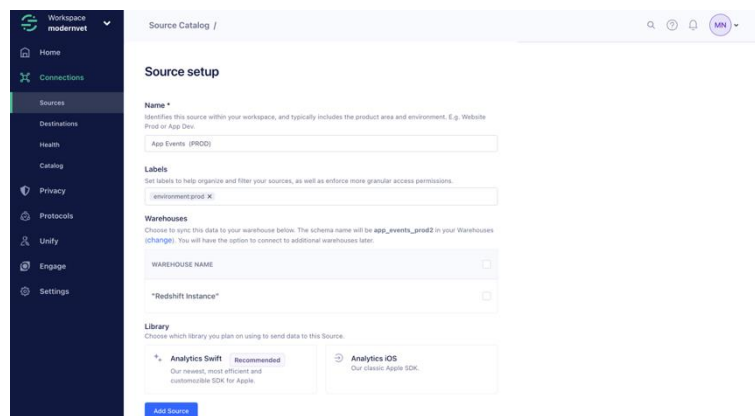
2. *HTTP API*

Konfigurasi HTTP API memungkinkan pengambilan data dari sumber eksternal melalui protokol HTTP.

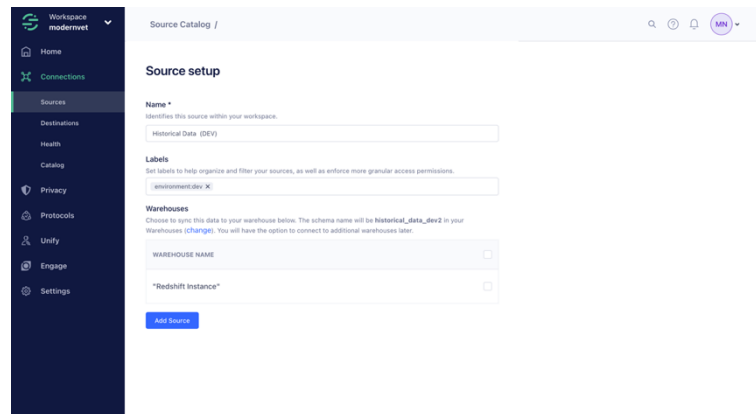
Kemudian langkah berikutnya adalah untuk melakukan konfigurasi sumber data *App Event* dan *HTTP API* pada *Dashboard Twilio Segment* untuk *Environment Development & Production* seperti pada gambar berikut:



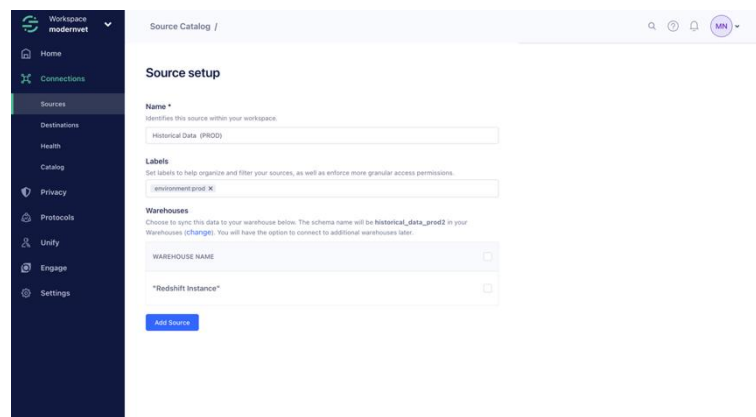
Gambar 3 Konfigurasi App Event Environment Development



Gambar 4 Konfigurasi App Event Environment Production



Gambar 5 Konfigurasi Historical Data Environment Development



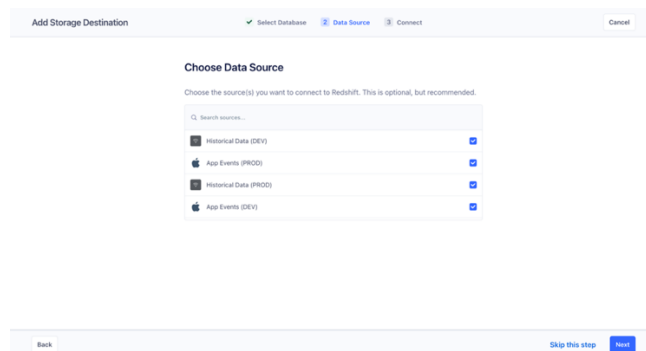
Gambar 6 Konfigurasi Historical Data Environment Production

Sebagai hasilnya, telah tersedia total 4 sumber data pada segment:

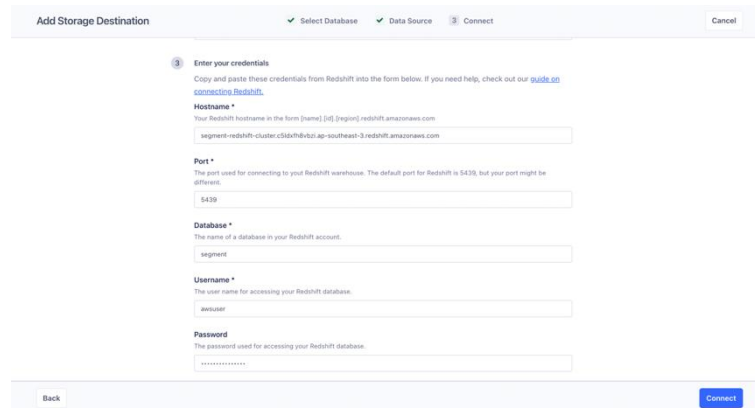
- 2x Apple: App Event Dev & App Event Prod
- 2x HTTP API: Historical Data Dev & Historical Data Prod

b. Konfigurasi Destinasi Data

Langkah berikutnya adalah mengarahkan data yang telah di kumpulkan oleh Twilio Segment ke destinasi yang sesuai. Pada tahap ini, fokus utama adalah mengintegrasikan Twilio Segment dengan Amazon Redshift sebagai destinasi data. Berikut adalah untuk melakukan konfigurasi destinasi data seperti gambar di bawah:



Gambar 7 Memilih Sumber Data



Gambar 8 Menghubungkan Redshift



Gambar 9 Hasil Konfigurasi Sumber Data & Destinasi Data

Berdasarkan gambar di atas, dapat dilihat bahwa Twilio Segment menyederhanakan proses pengiriman data ke destinasi. Selanjutnya yang perlu dilakukan adalah memastikan data disinkronkan dengan segment.

2. Integrasi Twilio Segment Dengan Sumber Data

Menghubungkan sumber data adalah langkah yang memungkinkan aplikasi Modernvet untuk secara efisien mengumpulkan dan mengelola data dari berbagai sumber, termasuk event tracking dalam aplikasi, API objek, dan database internal.

Twilio Segment memiliki beberapa cara untuk menghubungkan sumber data. Berikut adalah cara yang digunakan pada proses ini:

1. Segment Event Tracking

Cara ini digunakan untuk memposting acara ke Twilio Segment. Penggunaan paket flutter_segment sebagai solusi platform hybrid untuk memposting acara ke Twilio Segment.

2. Segment Object API

Memungkinkan pengambilan data dari sumber eksternal melalui protokol HTTP. Penggunaan API ini adalah untuk memastikan semua sumber data selalu terbaru dan selaras dengan Twilio Segment.

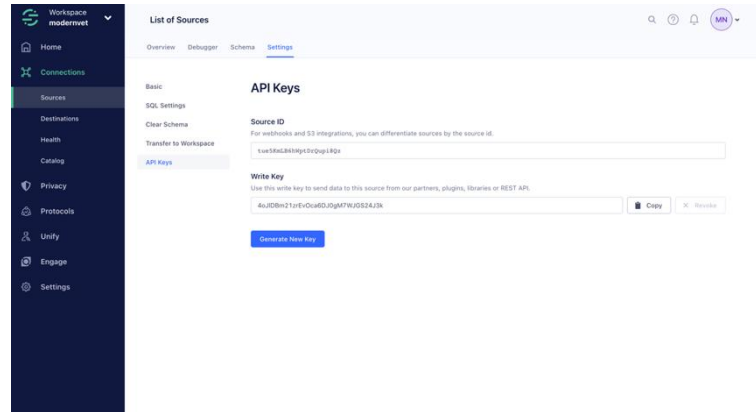
a. Menghubungkan Sumber Data In-App Events

Langkah pertama yang perlu dilakukan untuk menghubungkan sumber data in-app events adalah instalasi package flutter_segment pada file pubspec.yaml ke dalam proyek aplikasi modernvet. Pada proyek aplikasi tersebut menggunakan framework flutter:

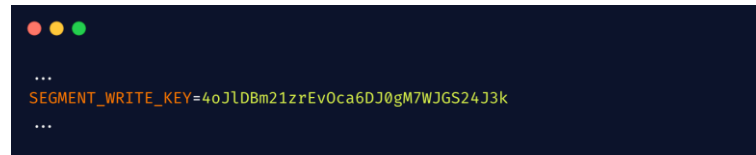
```
dependencies:
  flutter_segment: ^3.13.1
```

Gambar 10 Install Flutter Segment

Selanjutnya memasukan writeKey dari konfigurasi sebelumnya pada dashboard twilio segment ke dalam file .env pada projek aplikasi modernvet:



Gambar 11 Write Key



Gambar 12 Menambahkan Write Key Pada Env

Langkah selanjutnya adalah inisialisasi flutter segment pada projek aplikasimodernvet:



Gambar 13 Inisialisasi Flutter Segment

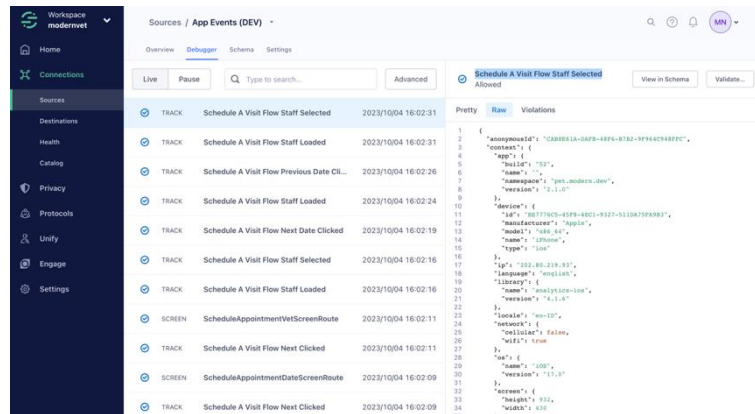
Langkah terakhir adalah menambahkan pelacakan event pada projek aplikasi Modernvet. Hasil dari pelacakan tersebut akan di kirim dan di ditampilkan pada dashboard twilio segment. Pada gambar di bawah menunjukan salah satu pelacakan yang akan di kirim ke twilio segment:

```

1 void onSelectVet(Staff? selectedVet) {
2   Segment.track(
3     eventName: 'Schedule A Visit Flow Staff Selected',
4     properties: {
5       'pets': selectedPets.map((p) =>
6         p.getPetEventProperties()).toList(),
7       'staff': selectedVet?.getStaffEventProperties(),
8     });
9   emit(state.copyWith(selectedVet: selectedVet));
10 }

```

Gambar 14 Contoh Pelacakan Event Pada Aplikasi



Gambar 15 Contoh Debugger App Event Pada Dashboard Segment

b. Menghubungkan Sumber Data dari Vetlink & Database

Untuk memastikan bahwa semua sumber data selalu terbaru dan selaras dengan segment penulis menggunakan AWS lambda, layanan komputasi tanpa server yang berjalan secara otomatis setiap hari pukul 19:00 untuk meminimalkan dampak kinerja potensial pada pengguna. Setiap kali berjalan, lambda menyinkronkan semua sumber data dari vetlink dan database ke segment melalui Segment Object API.

Lambda yang dibuat menggunakan SAM CLI penulis beri nama dengan Segment-Data-Worker yang akan menjalankan tugas-tugas berikut:

1. Pengambilan Data dari Vetlink dan Database
 Fungsi ini akan mengambil data dari sumber data Vetlink dan database. Data yang diambil kemudian akan diproses dan diteruskan ke Twilio Segment menggunakan Segment Object API. Dengan cara ini, penulis memastikan bahwa data yang diperlukan selalu tersedia di Twilio Segment untuk keperluan analisis lebih lanjut.
2. Penggunaan AWS Eventbridge
 Penulis menggunakan AWS Eventbridge untuk menjadwalkan trigger fungsi Lambda Segment-Data-Worker. Fungsi ini dijadwalkan untuk dijalankan secara otomatis setiap harinya pada pukul 19:00. Hal ini dilakukan agar data pelanggan selalu terkini.
3. Pencatatan Keluaran ke AWS CloudWatch
 Seluruh log dari fungsi ini akan dicatat dan dipantau melalui layanan AWS

CloudWatch. Langkah ini penting untuk memantau integrasi secara berkala.

Twilio Segment akan menjalankan sinkronisasi data dua kali sehari ke dalam destinasi Redshift. Tindakan ini bertujuan agar data yang telah penulis kumpulkan di Twilio Segment tersedia di dalam destinasi Redshift, yang dapat digunakan untuk analisis yang lebih mendalam.

Berikut ini adalah salah satu contoh penarikan data dari database dan penarikan data dari vetlink:

```

1 const appointmentsToFetchDate =
2   jobData[
3     getSuccessfulJobDataEntryFromSourceName(
4       JobSourceName.VETLINK_APPOINTMENTS,
5     )
6   ].slice();
7 await fetchAndSendDataInPages(
8   async (page, pageSize) =>
9     logAndFetchData(
10      JobSourceName.VETLINK_APPOINTMENTS,
11      getPreviousDayAndFutureVetlinkAppointments,
12      [appointmentsToFetchDate, page, pageSize],
13    ),
14   async vetlinkAppointments => {
15     await processDataAndSendToSegment(
16       JobSourceName.VETLINK_APPOINTMENTS,
17       vetlinkAppointments,
18       batchPostObjectsToSegment,
19       [
20         SegmentCollectionName.VETLINK_APPOINTMENTS,
21         vetlinkAppointmentsToSegmentAppointments,
22       ],
23       jobData,
24     );
25   },
26 );

```

Gambar 16 Penarikan Data Appointment Dari Vetlink

```

1 const dbXenditPaymentsToFetchDate = new Date(
2   jobData[
3     getSuccessfulJobDataEntryFromSourceName(JobSourceName.DB_XENDIT_PAYMENT
4     S)
5   ],
6 );
7 await fetchAndSendDataInPages(
8   async (page, pageSize) =>
9     logAndFetchData(
10      JobSourceName.DB_XENDIT_PAYMENTS,
11      getAllDBXenditPaymentsByUpdatedAt,
12      [dbXenditPaymentsToFetchDate, page, pageSize],
13    ),
14   async dbXenditPayments => {
15     await processDataAndSendToSegment(
16       JobSourceName.DB_XENDIT_PAYMENTS,
17       dbXenditPayments,
18       batchPostObjectsToSegment,
19       [
20         SegmentCollectionName.DB_XENDIT_PAYMENTS,
21         dbXenditPaymentsToSegmentDBXenditPayments,
22       ],
23       jobData,
24     );
25   },
26   100,
27 );

```

Gambar 17 Penarikan Data Pembayaran Dari Database

Kemudian berikut ini adalah proses yang dilakukan untuk melakukan pengiriman data dari *database* dan *vetlink* ke *segment* menggunakan *Segment Object API*:

```

1  const SEGMENT_SET_PATH = 'v1/set';
2  const SEGMENT_BATCH_SIZE = 10;
3
4  const segmentInstance = axios.create({
5    baseURL: process.env.SEGMENT_OBJECT_URL,
6    headers: {
7      'Content-Type': 'application/json',
8      Authorization: `Basic ${Buffer.from(
9        `${process.env.SEGMENT_WRITE_KEY}:`,
10       ).toString('base64')}`,
11    },
12  });
13
14  export const postToSegment = (objects, collection) =>
15    segmentInstance.post(SEGMENT_SET_PATH, { collection, objects });
16
17  export const batchPostObjectsToSegment = async (
18    objects,
19    collection,
20    objectMapper,
21  ) => {
22    for (let i = 0; i < objects.length; i += SEGMENT_BATCH_SIZE) {
23      const segmentObjects = objectMapper(
24        objects.slice(i, i + SEGMENT_BATCH_SIZE),
25      );
26      await postToSegment(segmentObjects, collection)
27        .catch(async () => {
28          // Retry once then throw the error if it still fails
29          await postToSegment(segmentObjects, collection).catch(error => {
30            throw error;
31          });
32        })
33        .catch(error => {
34          throw error;
35        });
36    }
37  };
38

```

Gambar 18 Pengiriman Data Ke Twilio Segment

Berikut ini adalah konfigurasi untuk Penggunaan AWS Eventbridge dan AWS CloudWatch pada file *template.yaml*:

```

1  Events:
2    CloudWatchEvent:
3      Type: Schedule
4      Properties:
5        Schedule: cron(0 19 * * ? *)
6

```

Gambar 19 Konfigurasi AWS Eventbridge & AWS Cloudwatch

Dan berikut ini adalah script untuk menjalankan fungsi lambda *segment-data-worker* tersebut dan juga contoh tampilan log dari hasil pengiriman data ke *segment*:

```

1 "scripts": {
2   "build": "sam build 66 npx prisma generate",
3   "test": "npm run build 66 npx jest",
4   "format": "eslint src --ext \\.mjs\\ --fix",
5   "lint": "eslint src --ext \\.mjs\\",
6   "start": "npm run build 66 sam local invoke --env-vars
  env.json",
7   "start:prod": "npm run build 66 sam local invoke --env-vars
  env-prod.json"
8 },
9

```

Gambar 20 Script Menjalankan Fungsi Lambda Segment Data Worker

```

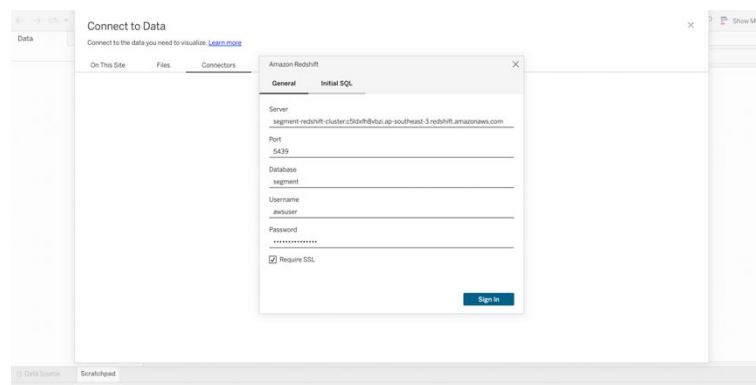
2023-10-16T08:17:39.091Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 1
2023-10-16T08:17:39.092Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Posting 19 bills to vetlink_invoices
2023-10-16T08:17:41.343Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Posting 28 pets to vetlink_subjects
2023-10-16T08:17:43.571Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Posting 33 pet items to vetlink_items
2023-10-16T08:17:47.552Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Finished posting 19 VetLink Invoices to segment
2023-10-16T08:17:48.298Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    ----- PROCESS VETLINK INVOICES -----
2023-10-16T08:17:48.298Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 2
2023-10-16T08:17:48.281Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    ----- PROCESS VETLINK APPOINTMENTS -----
2023-10-16T08:17:49.397Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    VetLink Appointments data count: 54
2023-10-16T08:17:49.408Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 1
2023-10-16T08:17:50.641Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Finished posting 54 VetLink Appointments to segment
2023-10-16T08:17:55.492Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    ----- PROCESS VETLINK APPOINTMENTS -----
2023-10-16T08:17:56.643Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    VetLink Appointments data count: 50
2023-10-16T08:17:56.643Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 2
2023-10-16T08:18:00.783Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Finished posting 50 VetLink Appointments to segment
2023-10-16T08:18:00.785Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    ----- PROCESS VETLINK APPOINTMENTS -----
2023-10-16T08:18:01.782Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    VetLink Appointments data count: 50
2023-10-16T08:18:01.783Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 3
2023-10-16T08:18:06.936Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Finished posting 50 VetLink Appointments to segment
2023-10-16T08:18:06.938Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    ----- PROCESS VETLINK APPOINTMENTS -----
2023-10-16T08:18:07.729Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 4
2023-10-16T08:18:07.733Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    ----- PROCESS VETLINK CLIENTS -----
2023-10-16T08:18:08.054Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    VetLink Clients data count: 9
2023-10-16T08:18:08.852Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 1
2023-10-16T08:18:09.872Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Finished posting 9 VetLink Clients to segment
2023-10-16T08:18:09.872Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    ----- PROCESS VETLINK CLIENTS -----
2023-10-16T08:18:10.489Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 2
2023-10-16T08:18:10.632Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    ----- PROCESS VETLINK PETS -----
2023-10-16T08:18:11.207Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    VetLink Pets data count: 10
2023-10-16T08:18:11.208Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Fetched page: 1
2023-10-16T08:18:11.946Z    a5480033-ed89-4fde-b2f-2681f56c648    INFO    Finished posting 10 VetLink Pets to segment

```

Gambar 21 Log Pengiriman Data Ke Twilio Segment

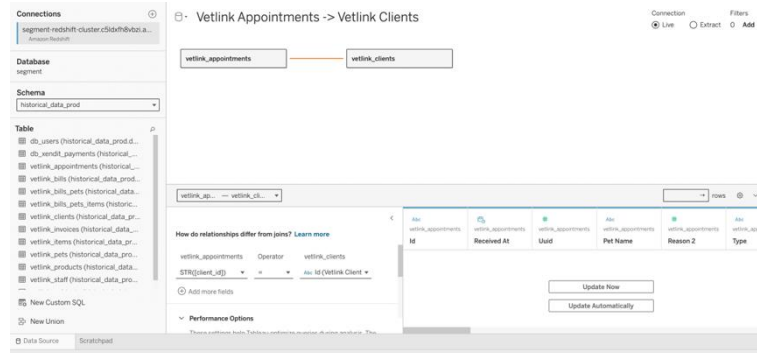
c. Menghubungkan Destinasi Data Dengan Tableau

Pada proses ini destinasi data yang di gunakan adalah redshift maka untuk menghubungkan destinasi data dengan tableau cukup dengan meng-import seperti berikut ini:



Gambar 22 Import Redshift Ke Tableau

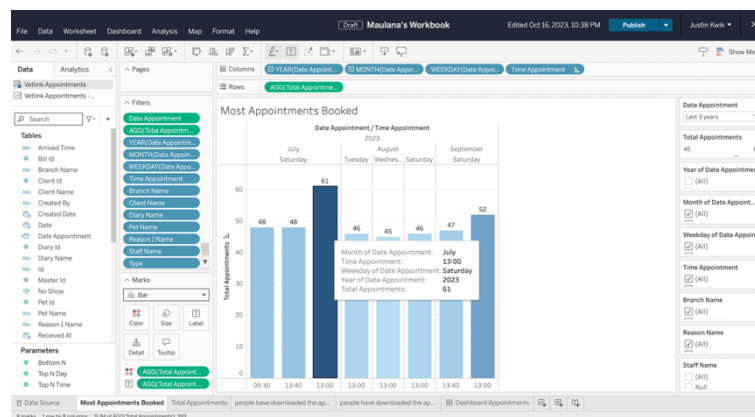
Proses selanjutnya akan dijelaskan contoh menampilkan salah satu visualisasi data yaitu menampilkan visualisasi “hari dan waktu dengan janji temu terbanyak”, maka penulis akan membuat Data Source dari schema historical_data_prod dan menggabungkan 2 tabel yaitu vetlink_appointments dengan vetlink_clients seperti pada gambar berikut:



Gambar 23 Membuat Data Source Di Tableau

3. Menampilkan Visualisasi Data

Gambar dibawah ini adalah salah satu contoh menampilkan visualisasi data menggunakan data source yang sudah di buat sebelumnya yaitu Vetlink Appointment -> Vetlink Client untuk menampilkan hasil hari dan waktu dengan janji temu terbanyak:



Gambar 24 Visualisasi Hari Dan Waktu Dengan Janji Temu Terbanyak

Berikut adalah cara – cara yang dilakukan pada proses visualisasi data tersebut:

1. Menambahkan data source vetlink_appointments
2. Menduplikasi data Date pada tabel vetlink_appointment dan merubah tipe datanya menjadi Date dan di beri nama dengan Date Appointment kemudian di tambahkan pada bagian Filter
3. Menduplikasi data Schedule Start Time pada tabel vetlink_appointment dan merubah isi data nya menjadi format jam dan menit dengan formula LEFT([Scheduled Start Time], 5) dan di beri nama Time Appointment
4. Membuat Calculated Field untuk mencari total data janji temu atau appointment berdasarkan id appointment dengan formula COUNT([Id]) dan di beri nama Total Appointment
5. Menambahkan 3 data Date Appointment pada bagian Columns dan Filter dengan format yang berbeda yaitu tahun, bulan dan hari
6. Menambahkan data Time Appointment pada bagian Columns dan Filter
7. Manambahkan data Total Appointment pada bagian Rows, Color, Label dan Filter

8. Menambahkan data lainnya ke dalam Filter seperti Branch Name, Client Name, Diary Name, Pet Name, Reason 1 Name, Staff Name, dan Type.

Kesimpulan

Kesimpulan - kesimpulan utama yang dapat diambil dari penelitian sebagai berikut:

1. Integrasi Twilio Segment sebagai pengelola data pelanggan Modernvet dari berbagai sumber dan jenis data berhasil di implementasikan.
2. Penggunaan Tableau untuk visualisasi data pelanggan memberikan wawasan bagi tim manajemen dalam pengambilan keputusan.

Daftar Pustaka

- Sawitri, D. (2019). *REVOLUSI INDUSTRI 4.0: BIG DATA MENJAWAB TANTANGAN REVOLUSI INDUSTRI 4.0*.
- Septa, F., & Alfia, D. (n.d.). *Literature Review Visualisasi Data dan Sistem Informasi Geografis Literature Review Visualisasi Data dan Sistem Informasi Geografis Literature Review Data Visualization and Geographic Information Systems 1)* Famia Septa Dinda Alfia, 2) Agussalim*.
<https://doi.org/10.36418/comserva.v2i08.493>
- * S. W., Nurharsono, T., Raharjo, A., Pendidikan, J., Kesehatan, J., Rekreasi, D., & Keolahragaan, I. (2013). **PEMBINAAN PRESTASI EKSTRAKURIKULER OLAHRAGA DI SMA KARANGTURI KOTA SEMARANG** Info Artikel _____ Sejarah Artikel: Diterima Januari 2013 Disetujui Februari 2013 Dipublikasikan Agustus 2013. In *Journal of Physical Education, Sport, Health and Recreation* (Vol. 2, Issue 8).
<http://journal.unnes.ac.id/sju/index.php/peshr>