

Integrasi Labeling Customer menggunakan WAHA dan n8n



n8n

Dokumentasi Teknis Tim *Acquisition*

Impacta
Digital Marketing Agency and Consultant

1. Latar Belakang

WhatsApp merupakan kanal utama untuk berinteraksi dengan calon pelanggan. Setiap percakapan yang terjadi antara *customer service* (CS) dan calon pelanggan mengandung informasi penting yang dapat digunakan untuk memahami kebutuhan, minat, ataupun keresahan calon pelanggan dalam proses penawaran. Calon pelanggan dikelompokkan menggunakan label pada setiap kontak berdasarkan kebutuhan, minat, ataupun keresahannya. Label tersebut mencerminkan segmentasi grading calon customer, misalnya :

- “**Grade A**” yang berarti calon customer siap mendapatkan penawaran dari customer service, atau
- “**Grade E**” yang berarti customer belum siap mendapatkan penawaran dari customer service.

Segmentasi ini sangat krusial karena akan menentukan strategi komunikasi berikutnya, baik dalam bentuk *follow up* manual, *broadcast* pesan, hingga kampanye remarketing lainnya. Sebelum dibuatkan tools ini, sistem labeling menggunakan sistem manual. Label yang sudah ada di aplikasi *WhatsApp mobile* akan diekspor dan dicocokkan secara manual dengan database calon pelanggan. Proses ini cukup melelahkan dan kurang efektif. Oleh karena itu, dibutuhkan sistem yang dapat mencatat aktivitas penambahan dan penghapusan label secara otomatis dan *realtime*, serta penyimpanan dalam *database* untuk keperluan analisis lanjutan.

Sebagai solusi, digunakan integrasi antara **WAHA (WhatsApp HTTP API)** dan **n8n** untuk membangun alur kerja (*workflow*) otomatis yang menerima *webhook* dari **WAHA** ketika ada perubahan label, memproses data tersebut, menyimpannya ke database *MySQL* dimana data calon customer berada. Dengan tools ini, proses pelabelan menjadi lebih efektif dan efisien, serta terdokumentasi dengan baik dari segi perubahan label untuk kebutuhan analisis jangka panjang.

2. Tujuan Dokumentasi

Dokumentasi ini disusun untuk memberikan pemahaman menyeluruh mengenai sistem pelabelan pelanggan menggunakan *WAHA* dan *n8n*. tujuan utama dari dokumentasi ini meliputi :

1. **Memperkenalkan Sistem *WAHA* (*WhatsApp HTTP API*)**

Memberikan gambaran umum mengenai *WAHA* sebagai solusi *API WhatsApp* yang digunakan untuk memberikan data label. Dokumentasi ini menjelaskan cara *WAHA* berkomunikasi dengan sistem *n8n*.

2. **Menjelaskan Alur Kerja Sistem (*Workflow*)**

Memberikan penjelasan detail mengenai bagaimana sistem menerima *webhook* dari *WAHA*, memproses data, dan menyimpannya ke dalam database secara otomatis melalui *n8n*.

3. **Memberikan Panduan Implementasi**

Memberikan referensi teknis bagi tim developer dalam memahami cara kerja *WAHA*, *n8n*, struktur *node*, logika pemrosesan data, serta konfigurasi lainnya.

4. **Mendukung Pemeliharaan dan Pengembangan**

Mempermudah proses *troubleshooting*, *debugging*, dan pengembangan lebih lanjut melalui dokumentasi struktur sistem dan fungsi tiap *node* yang digunakan.

3. Teknologi yang digunakan

Sistem ini dibangun menggunakan beberapa teknologi utama yang saling mendukung dan terintegrasi dalam proses otomatisasi pelabelan pelanggan dari *WhatsApp* ke basis data internal calon pelanggan. Teknologi yang kami gunakan adalah sebagai berikut :

3.1 WAHA (*WhatsApp HTTP API*)

adalah *self hosted WhatsApp API* yang memungkinkan integrasi aplikasi pihak ketiga dengan *WhatsApp Business* melalui *HTTP*. Dalam sistem ini *WAHA* digunakan untuk :

- Menerima dan menangani aktivitas *WhatsApp* secara real time.
- mengirimkan *webhook* saat terjadi penambahan atau penghapusan label pada kontak pelanggan.
- Menyediakan data *JSON* terstruktur yang dapat diproses oleh sistem *n8n*.

3.2 *n8n* (*Node based Workflow Automation Tool*)

adalah *platform automasi* berbasis *node* yang memungkinkan integrasi aplikasi satu dengan aplikasi lainnya melalui *workflow* visual. *n8n* berperan sebagai :

- Penerima *webhook* dari *WAHA*.
- Pemrosesan data untuk menyiapkan data yang diperlukan.
- Pengiriman data akhir ke database calon pelanggan.
- Pengaturan logika alur kerja (*conditional branching*) untuk membedakan antara peristiwa label ditambah dan label dihapus.

3.3 *MySQL Database*

adalah *platform* penyimpanan data berisi data pelanggan (database *CRM*), peran utamanya adalah :

- Menyimpan informasi terkait label yang ditambahkan atau dihapus oleh CS.
- Menyediakan basis data historis yang dapat digunakan untuk keperluan monitoring, analisis, dan pelaporan.
- Menjamin konsistensi data melalui operasi *upsert* (*insert* atau *update* berdasarkan *timestamp_unix*).

3.4 Metabase

adalah *platform* visualisasi data sederhana yang digunakan untuk visual data CRM, detailnya sebagai berikut :

- Melihat daftar calon pelanggan beserta label terakhir yang masih aktif berdasarkan peristiwa tambah label dari WAHA.
- Menganalisis informasi label dengan data marketing seperti sumber trafik (*utm source*, *utm campaign*, dlsb), *status lead*, respon, serta indikator lainnya.

4. Alur Kerja Sistem

Sistem ini dirancang untuk secara otomatis mencatat aktivitas pelabelan pelanggan di *WhatsApp* melalui integrasi antara *WAHA* dan *n8n*, serta menyimpannya ke dalam database *MySQL*. Alur kerja ini memastikan setiap perubahan label terekam dengan baik untuk keperluan analisis dan evaluasi performa CS.

4.1 Diagram alur sistem

WhatsApp Labeling → *WAHA* → *Webhook (WAHA Trigger)* → *n8n* → *Proses Filtering & Mapping* → *Simpan ke Database MySQL* → *Visualisasi di Metabase*

4.2 Penjelasan Alur

1. *Labeling* oleh CS di *WhatsApp*

CS memberikan label ke kontak calon customer melalui aplikasi *WhatsApp Business*. Label ini merepresentasikan segmentasi (misalnya “*Grade A*”, “*Grade E*”, dsb).

2. *WAHA* Mengirim *Webhook*

Saat label ditambahkan atau dihapus, *WAHA* secara otomatis mengirimkan *webhook HTTP* ke *endpoint n8n*. *Payload* berisi informasi label, nomor kontak, waktu, dan status (*add_label* / *delete_label*).

3. *n8n* Menerima dan Memproses Data

Webhook diterima oleh node trigger di *n8n*. Data kemudian difilter, diformat, dan diolah untuk dibedakan antara penambahan atau penghapusan label.

4. Data Disimpan ke *MySQL*

Informasi hasil pemrosesan dikirim ke tabel *labeling_hansarang_cs1* menggunakan metode *upsert* (*insert/update* berdasarkan *timestamp_unix*).

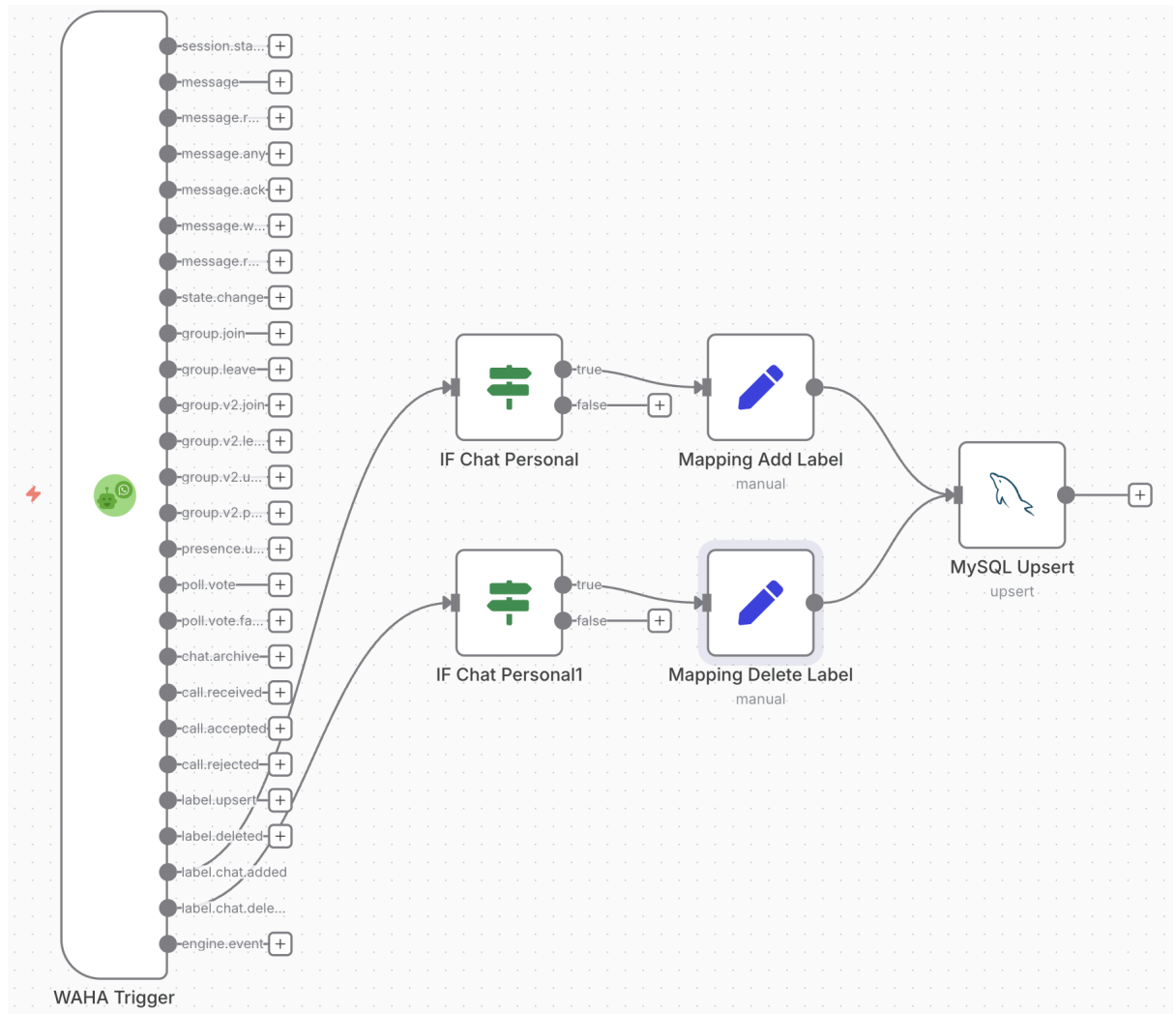
5. Visualisasi di *Metabase*

Data label aktif yang tersimpan dapat divisualisasikan di *Metabase* untuk analisis performa CS, segmentasi leads, dan evaluasi strategi komunikasi.

5. Detail Node

Bagian ini menjelaskan implementasi teknis dari alur kerja sistem di dalam *n8n*, mencakup struktur *node*, logika cabang, serta konfigurasi *field* yang digunakan.

5.1 Node utama dalam Workflow



<i>Node</i>	Fungsi
<i>WAHA Trigger</i>	Menerima <i>webhook</i> dari <i>WAHA</i> saat label ditambahkan/dihapus
<i>IF Chat Personal</i>	Mengecek apakah label ditambahkan (status = <i>add_label</i>)
<i>IF Chat Personal1</i>	Mengecek apakah label dihapus (status = <i>delete_label</i>)
<i>Mapping Add Label</i>	Menyiapkan <i>field JSON</i> untuk status <i>add_label</i>
<i>Mapping Delete Label</i>	Menyiapkan <i>field JSON</i> untuk status <i>delete_label</i>
<i>MySQL Upsert</i>	Menyimpan data label ke database <i>labeling_hansarang_cs1</i>

5.2 Konfigurasi *IF Chat* (Penting)

Node ini bernama *IF Chat Personal*, dan digunakan untuk memfilter jenis chat yang masuk melalui *webhook WAHA*, agar hanya memproses chat dari nomor personal, bukan dari grup atau channel. Berdasarkan standar *WhatsApp API* dan *WAHA*, *ID* pengguna *WhatsApp* memiliki struktur sebagai berikut:

Conditions

fx

{{ \$json.payload.chatId }}

AB ends with

@c.us

Add condition

Convert types where required

☐


Options

No properties

Add option

Tipe Chat	Suffix ID	Contoh
Chat Personal	@c.us	6281234567890@c.us
Grup	@g.us	120363025819748347@g.us
Broadcast	@broadcast	status@broadcast

5.3 Konfigurasi mapping data

 Mapping Add Label

Execute step

ParametersSettingsDocs

Mode

Manual Mapping

Fields to Set

timestamp_unix# Number

= {{ \$json.timestamp }}

klienAB String

= {{ \$json.session }}

nomor_csAB String

= {{ \$json.me.id.split('@')[0] }}

nomor_leadAB String

= {{ \$json.payload.chatId.split('@')[0] }}

label_idAB String

= {{ \$json.payload.label.id }}

statusAB String

add_label

payload.label.nameAB String

= {{ \$json.payload.label.name }}


label_nameAB String

= {{ \$json.payload.label.name }}

Drag input fields here or Add Field

Nama Field	Sumber Data	Keterangan
<i>timestamp_unix</i>	{{ \$json.timestamp }}	Waktu aktivitas labeling
<i>klien</i>	{{ \$json.session }}	Nama sesi/akun WAHA
<i>nomor_cs</i>	{{ \$json.me.id.split('@')[0] }}	Nomor WhatsApp CS
<i>nomor_lead</i>	{{ \$json.payload.chatId.split('@')[0] }}	Nomor WhatsApp calon customer
<i>label_id</i>	{{ \$json.payload.label.id }}	ID label dari WAHA
<i>label_name</i>	{{ \$json.payload.label.name }}	Nama label
<i>status</i>	'add_label' atau 'delete_label'	Status aktivitas labeling

5.3 Konfigurasi Node Database


MySQL Upsert
Execute step

Parameters
Settings
Docs

Credential to connect with

MySQL account

Operation

Insert or Update

Table

Name

labeling_hansarang_cs1

Data Mode

Map Each Column Below

Column to Match On

timestamp_unix

Used to find the correct row to update. Doesn't get changed. Has to be unique.

Value of Column to Match On

fx
{{ \$json.timestamp_unix }}

Values to Send

Values to Send

Column

klien

Value

`fx {{ $json.klien }}`

Column

nomor_cs

Value

`fx {{ $json.nomor_cs }}`

Column

nomor_lead

Value

`fx {{ $json.nomor_lead }}`

Column

label_id

Value

`fx {{ $json.label_id }}`

Column

status

Value

`fx {{ $json.status }}`

Column

label_name

Value

`fx {{ $json.payload.label.name }}`

Add Value

Node ini bertugas melakukan pemetaan dan transformasi data hasil dari *webhook WAHA* ke format yang sesuai untuk penyimpanan di database.

6. Struktur Table

Data hasil pelabelan yang diproses oleh *workflow n8n* disimpan secara terstruktur dalam sebuah tabel di database *MySQL* bernama *labeling_hansarang_cs1*. Tabel ini menjadi pusat penyimpanan histori aktivitas pelabelan yang dilakukan oleh tim *customer service*, baik dalam bentuk penambahan maupun penghapusan label.

6.1 Struktur Kolom

Nama Kolom	Tipe Data	Deskripsi
<i>timestamp_unix</i>	<i>BIGINT</i>	Waktu aktivitas dalam format <i>UNIX timestamp</i> (milidetik).
<i>klien</i>	<i>VARCHAR</i>	Nama sesi atau identifikasi dari client <i>WAHA</i> (<i>bot</i> atau <i>CS instance</i>).
<i>nomor_cs</i>	<i>VARCHAR</i>	Nomor <i>WhatsApp</i> milik <i>customer service</i> yang melakukan aktivitas.
<i>nomor_lead</i>	<i>VARCHAR</i>	Nomor <i>WhatsApp</i> calon pelanggan yang diberi atau dihapus labelnya.
<i>label_id</i>	<i>VARCHAR</i>	<i>ID</i> label yang diberikan oleh <i>WAHA</i> .
<i>label_name</i>	<i>VARCHAR</i>	Nama label yang digunakan (contoh: " <i>Grade A</i> ", " <i>Grade E</i> ", dsb).
<i>status</i>	<i>VARCHAR</i>	Menyimpan nilai ' <i>add_label</i> ' atau ' <i>delete_label</i> ' untuk membedakan jenis aktivitas.

6.2 Catatan Teknis

- Tabel ini bersifat *log table*, artinya menyimpan histori semua aktivitas labeling, bukan hanya status terakhir.

- Untuk keperluan analisis terakhir (*latest* label per nomor), digunakan teknik *window function* seperti *ROW_NUMBER()* untuk mengambil entri terbaru per kombinasi *nomor_lead* dan *label_id*.
- Penyimpanan *timestamp_unix* sebagai tipe *BIGINT* mempermudah konversi lintas zona waktu dan integrasi visualisasi seperti *Metabase*, *Looker Studio*, atau *Python dashboard*.

6.3 Contoh Data

<i>timestamp_unix</i>	klien	nomor_cs	nomor_lead	label_id	label_name	status
1753798945612	hansarang	6281234xxxx	6285678xxxx	1234	Grade A	add_label
1753798990000	hansarang	6281234xxxx	6285678xxxx	1234	Grade A	delete_label

7. Visualisasi Data di *Metabase*

Setelah data aktivitas pelabelan disimpan di tabel *labeling_hansarang_cs1*, data tersebut dapat divisualisasikan melalui *platform Metabase* untuk mendukung proses analisis, monitoring performa CS, serta evaluasi efektivitas label dalam segmentasi dan strategi *follow up*.

7.1 Tujuan Visualisasi

Visualisasi ini bertujuan untuk:

- Menampilkan daftar leads beserta label terakhir yang dimiliki.
- Menyambungkan data pelabelan dengan informasi detail leads (status, sumber, *campaign*, dsb).
- Mengelompokkan performa berdasarkan label (misalnya: konversi *Grade A vs Grade E*).
- Melihat tren penggunaan label dari waktu ke waktu.

7.2 CTE Latest Table

```
with latest_label as (  
  select *  
  from (  
    select  
      nomor_lead,  
      label_id,  
      label_name,  
      status,  
      timestamp_unix,  
      row_number() over (  
        partition by nomor_lead, label_id  
        order by timestamp_unix desc) as rn  
    from labeling_hansarang_cs1) t  
  where t.rn = 1 and t.status = 'add_label')
```


Keterangan Query

1. **ROW_NUMBER() OVER (...)** :

- Fungsi ini digunakan untuk mengurutkan baris per kombinasi *nomor_lead* dan *label_id* berdasarkan *timestamp_unix* terbaru (*desc*).
- *Partition by nomor_lead, label_id* artinya: setiap kombinasi *nomor_lead* dan *label_id* akan memiliki nomor urut sendiri. Tujuannya adalah kita ingin tahu label terakhir (*timestamp* terbaru) yang pernah diberikan ke setiap kontak (*nomor_lead*) untuk setiap label tertentu.

2. **WHERE t.rn = 1** :

- Mengambil hanya baris pertama (terbaru) dari masing-masing kombinasi *nomor_lead* dan *label_id*.

3. **AND t.status = 'add_label'** :

- Memastikan bahwa hanya label yang aktif (ditambahkan, bukan dihapus) yang diambil.
- Jika label tersebut sudah pernah dihapus (ada *delete_label* yang lebih baru), maka tidak akan muncul di hasil akhir.

8. Manfaat dan Penggunaan Data

Data pelabelan yang tersimpan secara otomatis di database *MySQL* memiliki manfaat strategis bagi tim operasional, marketing, maupun *data analyst*. Berikut beberapa poin utama manfaat dan implementasi penggunaan datanya :

8.1 Segmentasi dan Personalisasi Komunikasi

Dengan data label yang mencerminkan tingkat kesiapan atau minat calon pelanggan, tim CS maupun marketing dapat:

- Menyusun daftar kontak untuk *broadcast* berdasarkan label tertentu (misal: hanya “*Grade A*”).
- Mengatur intensitas dan jenis *follow up* berdasarkan prioritas label.
- Melakukan *remarketing* yang lebih tepat sasaran sesuai minat pelanggan.

8.2 Monitoring dan Evaluasi Strategi Marketing

Data label bisa menjadi indikator awal efektivitas kampanye. Contohnya:

- Jika iklan menghasilkan banyak lead “*Grade A*”, artinya pesan kampanye berhasil menjangkau audiens berkualitas.
- Jika banyak lead berlabel “*Grade E*” dari kanal tertentu, mungkin perlu evaluasi targeting iklan.

8.3 Integrasi Otomatisasi dan *Workflow* Lanjutan

Dengan data label yang tersimpan secara konsisten:

- Sistem dapat memicu otomatisasi lanjutan (misal: jika *customer* masuk *Grade A*, langsung kirim brosur atau katalog).
- Label dapat digunakan sebagai pemicu (*trigger*) di *CRM*, *email automation*, atau *WhatsApp blast system*.
- Mempermudah *filtering* pelanggan untuk proses distribusi *lead* ke tim CS atau *marketing* tertentu.

8.4 Pelacakan Riwayat Interaksi dan Audit

Pencatatan setiap perubahan label membantu untuk:

- Menelusuri histori aktivitas pelanggan: kapan diberi label, kapan dihapus.
- Memastikan transparansi dan akuntabilitas dalam proses pelabelan, terutama jika digunakan untuk menentukan prioritas atau nilai *business*.

8.5 Data *Insight* untuk *Forecasting & Decision Making*

Dengan menggabungkan data label dan data konversi, tim data dapat:

- Menentukan konversi per label (misal: *Grade A* → *30% closed*).
- Membuat model prediktif untuk *lead scoring* berbasis *behavior* pelabelan historis.
- Mendukung keputusan strategis seperti budget iklan, alokasi CS, atau perubahan *script* komunikasi.

9. Penutup dan Rencana Pengembangan

Implementasi sistem pelabelan otomatis menggunakan integrasi antara *WAHA (WhatsApp HTTP API)*, *n8n*, dan *database MySQL* telah memberikan solusi praktis dan efisien untuk mencatat aktivitas labeling pelanggan secara real time. Sistem ini tidak hanya membantu tim *Customer Service* dalam proses segmentasi dan *follow up*, tetapi juga menjadi fondasi data yang kuat untuk analisis perilaku pelanggan dan evaluasi efektivitas strategi pemasaran.

Dengan tercatatnya data label secara sistematis dan terdokumentasi, tim dapat :

- Menghindari proses manual yang rawan kesalahan,
- Menyusun strategi komunikasi berbasis data aktual,
- Memantau performa tim secara lebih objektif dan terukur,
- Serta mengintegrasikan data ini ke dalam ekosistem *CRM* dan visualisasi internal.

8.1 Rencana Pengembangan

Meskipun sistem pelabelan telah berjalan dengan baik dan digunakan secara aktif, terdapat peluang pengembangan lebih lanjut yang dapat meningkatkan pemanfaatan data secara strategis, khususnya dalam mendukung pengambilan keputusan marketing yang lebih tajam.

a. Visualisasi dan Analisis *Campaign* yang Lebih Mendalam

Saat ini, sistem telah menyimpan sumber asal pelanggan (*Google Ads*, *Meta Ads*, *SEO*, dll.) secara lengkap, namun visualisasi campaign belum optimal. Rencana ke depan adalah:

- Membuat dashboard segmentasi berdasarkan asal *campaign* dan label, contoh: berapa persen dari *traffic Meta Ads* yang masuk ke *Grade A*, atau *Grade E*.
- Analisis efektifitas iklan berbasis label, contoh: campaign mana yang menghasilkan *Grade A* terbanyak per 100 klik.

- Visualisasi performa per channel, per hari, atau per kategori produk.

Hal ini akan membantu tim marketing untuk:

- Mengevaluasi *return on ad spend (ROAS)* bukan hanya dari *conversion* akhir, tapi juga dari kualitas *lead* awal.
- Menyusun strategi pengalihan budget iklan ke channel yang lebih prospektif.

b. *Forecasting Campaign* Berdasarkan Data Historis

Untuk meningkatkan perencanaan ke depan, sistem dapat dikembangkan dengan model prediktif sederhana untuk:

- Memprediksi distribusi label dari campaign tertentu berdasarkan data historis (misal: *campaign* dengan *keyword* "X" biasanya menghasilkan 30% *Grade A*).
- Memproyeksikan potensi *closing* atau kebutuhan CS, sehingga tim bisa lebih siap saat *campaign* besar diluncurkan.
- Menyusun *early warning system* jika label yang masuk terlalu banyak *Grade E* atau tidak berkembang ke *Grade* lebih tinggi dalam waktu tertentu.

Pendekatan ini bisa dimulai dengan:

- Eksplorasi sederhana menggunakan *Python* dan visualisasi tren label mingguan.
- Lalu dilanjutkan dengan *model scoring* berbasis distribusi label dan data kampanye sebelumnya.