

INTEGRASI ODOO DENGAN WHATSAPPS

INTEGRASI ODOO DENGAN WHATSAPPS

Sumber :

1. Ctech ERP Consultant.
https://docs.google.com/document/d/1E7LCjSFBQ4wneeKSRfB2vZzKesQ1WD_WPyM_sZJOVV/edit?tab=t.0
2. https://www.odoo.com/id_ID/slides/slide/whatsapp-advanced-setup-5321
- 3.

Berdasarkan dokumen "**KUMPULAN PROMPT**"¹ yang Anda lampirkan, saya telah menganalisis dan memilihkan formula terbaik untuk masalah Anda. Karena Anda merasa "bingung dengan konfigurasi" (yang merupakan proses kompleks dan bertahap), rumus yang paling tepat digunakan adalah **R-I-S-E (Role - Input - Steps - Expectation)** atau **R-T-F (Role - Task - Format)**.

Namun, untuk panduan teknis yang lengkap seperti integrasi WhatsApp yang melibatkan pihak ketiga (Meta/Facebook), saya merekomendasikan **Gabungan R-I-S-E dan Context** agar AI memahami apa yang sudah Anda pelajari.

Berikut adalah **Prompt Final Terbaik** yang saya buat khusus untuk Anda berdasarkan rumus di dokumen tersebut:

Prompt Final: Panduan Konfigurasi Odoo - WhatsApp

Silakan copy-paste prompt di bawah ini ke Gemini/ChatGPT:

Role (Peran):

Bertindaklah sebagai Senior Odoo Technical Consultant dan Meta API Specialist yang berpengalaman 10 tahun dalam integrasi ERP. Anda ahli dalam menjembatani dokumentasi teknis Odoo dengan konfigurasi Meta for Developers.

Input (Konteks & Masalah):

Saya adalah konsultan ERP di PT. Ctech ERP Indonesia. Saat ini saya sedang melakukan konfigurasi koneksi Odoo dengan WhatsApp (Native Integration).

Saya sudah mempelajari panduan dari:

1. Website Teguhteja (Complete Configuration Guide)

<https://teguhteja.id/whatsapp-integration-with-odoo-complete-configuration-guide/>

2. Video Youtube :

- <https://www.youtube.com/watch?v=BZcuqWhk4Z0>
- <https://www.youtube.com/watch?v=Y45ZZSH8j1s>
- <https://www.youtube.com/watch?v=xtjS8e7OPOE>
- <https://www.youtube.com/watch?v=IGiOEAhPyNE>
-

Namun, saya masih bingung dan mengalami kendala dalam menghubungkan kedua platform tersebut secara benar. Saya membutuhkan panduan yang meluruskan pemahaman saya dari kedua sumber tersebut.

Steps (Langkah Pengerjaan):

1. **Meta Setup:** Jelaskan langkah kunci di dashboard *Meta for Developers* (pembuatan App, konfigurasi produk WhatsApp, dan pengambilan Token/Phone Number ID) yang sering terlewat.
2. **Odoo Configuration:** Pandu langkah konfigurasi di menu *Settings > General Settings* Odoo (IAP vs Custom Account) dan cara memasukkan kredensial yang benar.
3. **Webhook Verification:** Jelaskan cara setting Webhook Verify Token agar status pesan (Sent/Read) bisa masuk kembali ke Odoo.
4. **Troubleshooting:** Berikan 3 kesalahan umum (common pitfalls) yang sering membuat koneksi gagal/terputus dan cara mengatasinya.

Expectation (Hasil yang Diharapkan):

Buatkan panduan langkah demi langkah (Step-by-Step Guide) yang terstruktur, menggunakan bahasa Indonesia yang lugas, lengkap dengan nama menu yang presisi. Jika ada perbedaan antara Odoo Community dan Enterprise dalam hal ini, tolong jelaskan.



Penjelasan Mengapa Ini Prompt Terbaik (Berdasarkan Sumber Dokumen)

Prompt di atas disusun menggunakan prinsip-prinsip dari dokumen Anda:

1. R - Role (Peran): 2
Saya menetapkan peran spesifik "Senior Odoo Technical Consultant & Meta API Specialist" agar jawaban tidak hanya umum, tetapi menyentuh aspek teknis API yang sering menjadi sumber kebingungan.
2. I - Input (Masukan): 3
Sesuai tips "Berikan Konteks" di Bab 1 dokumen, saya memasukkan referensi (Teguhteja & Youtube) agar AI tahu "Knowledge Base" Anda saat ini dan bisa mengisi celah informasi yang mungkin hilang dari sumber tersebut.

3. S - Steps (Langkah): 4

Menggunakan rumus R-I-S-E, saya memecah instruksi menjadi 4 langkah logis (Meta Setup -> Odoo Config -> Webhook -> Troubleshooting). Ini mencegah AI memberikan jawaban yang loncat-loncat.

4. E - Expectation (Ekspektasi): 5

Saya meminta format "Step-by-Step Guide" yang terstruktur, sesuai dengan tips dokumen untuk "Tentukan Format Output".

Gunakan prompt ini, dan Anda akan mendapatkan panduan yang jauh lebih rapi daripada sekadar bertanya "Cara setting Odoo WA bagaimana?". Selamat mencoba!

Tentu, ini adalah panduan lengkap yang Anda butuhkan. Sebagai Senior Consultant yang sering menangani kasus ini, saya akan meluruskan kebingungan Anda, terutama mengenai perbedaan antara *Odoo IAP* (jasa broker Odoo) dan *Native Connection* (koneksi langsung ke Meta) yang sedang Anda coba.

Banyak tutorial di YouTube (termasuk beberapa link yang Anda kirim) sering melompati bagian "Token Permanen", yang menyebabkan koneksi putus setelah 24 jam.

Berikut adalah panduan **Step-by-Step** untuk Odoo 17 (dan 16 Enterprise) dengan fokus pada **Direct Connection (Custom Account)**.

Panduan Konfigurasi Odoo - WhatsApp (Native & Direct)

Asumsi Teknis:

- Anda menggunakan Odoo Enterprise (v16/v17+).
- Anda memiliki akses Admin di *Meta Business Manager & Odoo*.
- Nomor WhatsApp belum terdaftar di WA mobile app (nomor harus "bersih" atau dihapus akunnya dari HP).

1. Meta Setup: Dashboard Meta for Developers

Kunci keberhasilan ada di sini. Jangan hanya gunakan "Temporary Token".

1. Buat App:

- Masuk ke developers.facebook.com > **My Apps** > **Create App**.
- Pilih Type: **Other** > **Business**. (Penting: Jangan pilih "Consumer").
- Isi nama app (misal: "Odoo ERP") dan hubungkan dengan **Business Account** perusahaan Anda.

2. Setup WhatsApp Product:

- Di sidebar kiri, cari "WhatsApp" > klik **Set up**.
- **API Setup:** Anda akan melihat "Temporary Access Token". **JANGAN GUNAKAN**

INI UNTUK PRODUCTION. Token ini akan kadaluarsa dalam 24 jam.

3. Mendapatkan Permanent Token (Langkah Kritis yang Sering Terlewat):

- Buka **Business Settings** (di business.facebook.com) > **Users** > **System Users**.
- Klik **Add** > Beri nama "Odoo Bot" > Role: **Admin**.
- Klik **Generate New Token** > Pilih App yang tadi dibuat ("Odoo ERP").
- **Permissions (Wajib Checklist):**
 - whatsapp_business_messaging
 - whatsapp_business_management
- Copy token panjang yang muncul. Inilah **Access Token** abadi Anda.

4. Phone Number ID:

- Kembali ke menu **WhatsApp > API Setup** di Developer Console.
- Copy **Phone Number ID** (bukan nomor HP-nya, tapi ID angka panjang) dan **WhatsApp Business Account ID**.

2. Odoo Configuration: Settings & Credentials

Meluruskan IAP vs Custom Account.

1. Masuk Menu:

- Buka Odoo > Apps > Cari modul **WhatsApp** > Install.
- Buka app **WhatsApp > Configuration > WhatsApp Business Accounts**.
- Klik **New**.

2. Isi Kredensial:

- **Name:** Isi bebas (misal: "Kantor Pusat").
- **Account Provider:** Pilih **Meta** (Jika opsi ini tidak muncul dan langsung diminta beli kredit, Anda sedang di mode IAP. Pastikan developer mode aktif atau cek modul whatsapp teknisnya). *Catatan: Di Odoo 17 Enterprise standard, Odoo mengarahkan ke IAP. Untuk direct connection, Anda mengisi field API Credentials secara manual.*
- Masukkan data dari Langkah 1:
 - **App ID:** (Dari App Settings di Meta).
 - **App Secret:** (Dari App Settings > Basic di Meta).
 - **Access Token:** (Token System User dari langkah 1.3).
 - **Phone Number ID:** (Dari langkah 1.4).
 - **Account ID:** (Dari langkah 1.4).

3. Test Connection:

- Klik tombol **Test Connection**. Jika sukses, Odoo akan memunculkan pesan hijau dan otomatis mengenerate **Callback URL** dan **Verify Token**.

3. Webhook Verification: Agar Status "Sent/Read" Masuk Odoo

Tanpa ini, Anda bisa kirim pesan, tapi Odoo tidak tahu apakah pesan itu sudah dibaca atau gagal.

1. Ambil Data dari Odoo:

- Setelah *Test Connection* sukses, lihat field **Callback URL** dan **Verify Token** di form konfigurasi Odoo tadi. Copy keduanya.

2. Setting di Meta:

- Kembali ke Dashboard Meta Developer > **WhatsApp** > **Configuration**.
- Cari bagian **Webhook** > Klik **Edit**.
- Paste **Callback URL** dan **Verify Token** dari Odoo > Klik **Verify and Save**.

3. Subscribe to Events (Wajib):

- Setelah verify berhasil, klik **Manage** (di bagian Webhook fields).
- Centang (Subscribe) pada kolom berikut:
 - messages (untuk terima balasan chat).
 - message_deliveries (untuk status sent/delivered/failed).
 - message_reads (untuk status read/centang biru).
 - message_failures (opsional, untuk debug).

4. Troubleshooting: 3 Kesalahan Umum (Common Pitfalls)

1. Masalah: "Token Expired" atau Koneksi Putus Tiba-tiba

- *Penyebab*: Anda menggunakan *Temporary Access Token* dari halaman API Setup, bukan membuat *System User*.
- *Solusi*: Ulangi langkah 1.3 (System User) dan update token di Odoo.

2. Masalah: Pesan Tidak Terkirim (Gagal di Odoo)

- *Penyebab*: Anda mencoba mengirim pesan "Halo apa kabar?" (Free Text) sebagai pesan pertama.
- *Solusi*: Aturan Meta ketat. Pesan **pertama** (inisiasi bisnis) WAJIB menggunakan **Template** yang sudah diapprove. Anda tidak bisa memulai chat dengan teks bebas. Buat Template di Odoo > Kirim ke Meta untuk approval > Gunakan Template itu.

3. Masalah: Webhook Error / Status Tidak Update (Khusus Odoo On-Premise)

- *Penyebab*: Server Odoo Anda berada di balik Firewall atau menggunakan Localhost yang tidak bisa diakses publik oleh server Meta.
- *Solusi*: Pastikan URL Odoo Anda (<https://odoo.perusahaan.com>) dapat diakses dari internet publik. Jika masih development (lokal), gunakan *Ngrok* atau *Reverse Proxy* agar Meta bisa "menembak" Webhook ke Odoo Anda.

Semoga panduan ini membantu meluruskan "benang kusut" konfigurasi Anda. Silakan dicoba langkah demi langkah!

Sebagai sesama praktisi ERP, ini adalah analisis krusial yang harus Anda pahami sebelum merekomendasikan solusi ke klien.

Perbedaan mendasar dari ketiga tautan yang Anda berikan adalah **Jenis Jalur Koneksi (API**

Type).

1. **Meta (Official Cloud API):** Jalur resmi, "White Hat", stabil, namun berbayar per percakapan dan ketat aturannya.
2. **Watzap.id & Go4Whatsup:** Jalur tidak resmi (sering disebut "Grey API" atau "WA Gateway"), berbasis *reverse engineering* protokol WA Web. Lebih bebas, langganan *flat*, namun berisiko.

Berikut adalah **Analisis Perbandingan Mendalam** dalam tabel 5 kolom untuk membantu Anda memutuskan arsitektur di Odoo:

Tabel Perbandingan Solusi WhatsApp untuk Odoo

Kriteria / Aspek	1. Meta WhatsApp (Official API)	2. Watzap.id (Unofficial Gateway)	3. Go4Whatsup (Unofficial Gateway)	Implikasi Teknis di Odoo (ERP Context)
Metode Koneksi	Direct Cloud API. Server Odoo berbicara langsung ke Server Meta via HTTP Request. Tidak butuh HP yang menyala.	WA Web Wrapper. Sistem mereka meniru browser. Anda harus scan QR Code dari HP fisik yang harus selalu aktif/terkoneksi internet.	WA Web Wrapper. Sama seperti Watzap, menggunakan emulasi browser/socket untuk login via QR Code.	Meta: Native di Odoo 16/17 Ent. Tidak perlu modul 3rd party. Watzap/Go4: Wajib beli/coding modul <i>custom connector</i> (Add-on) untuk menjembatani Odoo ke API mereka.
Biaya (Cost)	Pay-per-Conversation. Bayar ke Meta per 24 jam sesi percakapan (Service/Utility/Marketing). 1000 service convos pertama gratis/bulan.	Langganan Flat. (Bulanan/Tahunan). Biasanya berkisar Rp 100rb - 300rb/bulan per nomor. Unlimited pesan (secara teori).	Langganan Flat. Mirip Watzap, model SaaS berlangganan dengan tiering berdasarkan fitur atau jumlah device.	Meta: Klien harus deposit kartu kredit di Meta Manager. Cost fluktuatif sesuai volume. Watzap/Go4: Cost tetap (Fixed), lebih mudah diprediksi untuk budget klien UKM.
Stabilitas & Risiko	Sangat Stabil. SLA terjamin oleh Meta.	Menengah. Tergantung koneksi	Menengah. Sama. Jika WA update	Meta: Cocok untuk notifikasi

	Zero Ban Risk selama mengikuti <i>Commerce Policy</i> .	HP fisik. High Ban Risk jika algoritma WA mendeteksi spam/blast berlebihan. Nomor bisa diblokir permanen.	algoritma/versi Web, layanan bisa down sementara sampai provider melakukan <i>patching</i> .	kritikal (Invoice, OTP, DO) yang <i>wajib sampai</i> . Watzap/Go4: Ada risiko pesan gagal terkirim jika HP mati atau nomor terblokir.
Aturan Pesan (Rules)	Ketat (Template Based). Memulai chat WAJIB pakai Template yang diapprove. Tidak bisa "Say Hi" sembarangan.	Bebas (Free Text). Bisa kirim pesan apapun kapanpun tanpa template. Bisa kirim pesan marketing bertubi-tubi.	Bebas (Free Text). Sama. Fleksibilitas konten sangat tinggi tanpa proses approval.	Meta: User Odoo harus bikin template di sistem dulu. Agak kaku untuk sales. Watzap/Go4: User Odoo bisa langsung kirim pesan custom dari Chatter/Log note tanpa batasan.
Status Centang (Green Tick)	Bisa (Verified Business). Memungkinkan dapat <i>Green Tick</i> jika brand terkenal/terverifikasi.	Tidak Bisa. Status di HP penerima tetap akun personal biasa (atau akun bisnis biasa tanpa centang hijau).	Tidak Bisa. Sama seperti Watzap.	Meta: Meningkatkan <i>Trust</i> brand klien Anda di mata konsumen. Watzap/Go4: Terlihat seperti nomor biasa.

Analisis Mendalam untuk Konsultan

Sebagai konsultan PT. Ctech ERP, Anda harus memposisikan ketiga opsi ini berdasarkan **Kebutuhan Klien:**

1. Kapan Memilih Meta (Official)?

Pilihlah ini jika klien adalah **Perusahaan Menengah-Besar (Corporate/Enterprise)** yang:

- Mementingkan *Brand Image* (ingin nama perusahaan muncul di header WA, bukan sekadar nomor HP).

- Membutuhkan **Reliability** tinggi untuk notifikasi transaksional (misal: "Pesanan Dikonfirmasi", "Faktur Jatuh Tempo").
- Menggunakan Odoo Enterprise versi terbaru (v16/v17) dan ingin meminimalisir penggunaan modul custom pihak ketiga.
- **Kelemahan:** Setup awal ribet (Verifikasi dokumen bisnis), dan berbayar per pesan.

2. Kapan Memilih Watzap.id / Go4Whatsup?

Pilihlah ini jika klien adalah **UMKM atau Tim Sales Agresif** yang:

- Ingin melakukan **Blast Marketing** (Promo) ke ribuan nomor tanpa takut biaya per pesan membengkak.
- Merasa ribet dengan aturan "Template Approval" Meta. Mereka ingin sales bisa chat bebas ke customer langsung dari Odoo.
- Budget terbatas dan ingin biaya flat (Fixed Cost).
- **Kelemahan:** Anda (Consultant) harus mencari atau membuat modul *connector* Odoo ke API Watzap/Go4Whatsup (tidak tersedia secara native). Dan Anda harus memperingatkan klien tentang risiko nomor diblokir (Banned) oleh WhatsApp.

Rekomendasi Prompt Selanjutnya

Jika Anda memutuskan lanjut dengan Meta (Official) sesuai panduan awal, Anda mungkin butuh template pesan.

"Buatkan saya contoh format Template WhatsApp (JSON) untuk notifikasi Sales Order Confirmation di Odoo, beserta variabel dynamic placeholder-nya."

Jika Anda ingin mencoba Watzap/Go4Whatsup:

"Bagaimana struktur coding Python dasar untuk memanggil API eksternal (POST Request) di Odoo saat tombol 'Confirm' pada Sales Order ditekan?"

Ini adalah temuan yang sangat relevan untuk pasar lokal Indonesia.

Berdasarkan analisis teknis dari dokumentasi API mereka (klikodoo.id/api/wa/...) dan struktur harganya, berikut adalah bedahan lengkap mengenai **KlikOdoo.id**:

Apa itu KlikOdoo.id?

KlikOdoo.id adalah penyedia layanan **WhatsApp Gateway (Unofficial/Grey API)** buatan lokal yang dikhususkan untuk ekosistem Odoo.

Secara teknis, KlikOdoo mirip dengan **Watzap** atau **Go4Whatsup**, tetapi dengan perbedaan kunci: mereka **fokus menyediakan Modul Odoo "Plug-and-Play"**. Jadi, Anda tidak perlu pusing coding API connector sendiri karena mereka menjual layanan *gateway* sekaligus modul

addon-nya.

Karakteristik Utama:

1. **Metode Koneksi:** Scan QR Code (Anda butuh HP fisik yang menyala dan terkoneksi internet).
2. **Sifat:** Unofficial (Bukan mitra resmi Meta). Menggunakan teknik *browser emulation* untuk meminjam sesi WA Web.
3. **Pricing Model:** Berlangganan bulanan dengan **Batasan Kuota Harian (Daily Limit)**.
 - *Small:* ~250 pesan/hari
 - *Large:* ~2.000 pesan/hari

Cara Menggunakannya (Workflow)

Jika Anda memilih menggunakan KlikOdoo untuk klien, alurnya jauh lebih sederhana dibanding Meta Official API:

1. **Registrasi & Langganan:**
 - Buka situs KlikOdoo, pilih paket (Small/Medium/Large) sesuai volume chat klien.
 - Lakukan pembayaran. Anda akan mendapatkan Client ID dan Client Secret.
2. **Instalasi Modul:**
 - KlikOdoo biasanya menyediakan file modul Odoo (zip) yang harus di-install ke server Odoo Anda (via folder custom-addons).
 - *Note:* Cek kompatibilitas versi Odoo (v14/v15/v16/v17) sebelum membeli.
3. **Konfigurasi di Odoo:**
 - Setelah install, masuk ke menu Settings modul tersebut.
 - Masukkan Client ID dan Client Secret yang didapat saat registrasi.
4. **Pairing Device (Scan QR):**
 - Di Odoo akan muncul tombol "Get QR Code".
 - Scan QR tersebut menggunakan HP yang ada WhatsApp-nya (Menu *Linked Devices* di HP).
 - Status akan berubah menjadi "Connected".
5. **Selesai:**
 - Anda bisa langsung kirim pesan dari SO/Invoice tanpa perlu membuat Template Approval ke Meta.

Analisis: KlikOdoo vs Meta Official (Untuk Konsultan)

Sebagai konsultan, Anda harus transparan kepada klien mengenai plus-minusnya:

Fitur	Meta Official (Native Odoo)	KlikOdoo.id (3rd Party)
Setup	Rumit (Verifikasi Bisnis, Token, System User).	Mudah & Cepat (Install Modul, Scan QR, Jalan).
Ketergantungan	Server Cloud (Tanpa HP Fisik).	HP Fisik. HP mati/internet

	Sangat stabil.	putus = Odoo gagal kirim WA.
Modul Odoo	Native (Bawaan Odoo Enterprise).	Wajib Install Addon. Jika Odoo update versi, Anda harus tunggu KlikOdoo update modulnya.
Limit Pesan	Tiering (1.000 - Unlimited per hari).	Dibatasi Paket (Misal max 2.000/hari). Tidak cocok untuk blast besar.
Keamanan	Aman dari blokir (Official).	Risiko Banned. Karena unofficial, jika blast terlalu agresif, nomor bisa diblokir WA.

💡 Kesimpulan & Rekomendasi

Gunakan KlikOdoo jika:

- Klien adalah UKM yang tidak punya kartu kredit korporat (untuk bayar Meta).
- Klien butuh fitur "Kirim pesan bebas" (tanpa template approval).
- Anda tidak ingin repot coding *connector* API sendiri (karena KlikOdoo sudah sediakan modulnya).

Tetap gunakan Meta Official jika:

- Klien adalah perusahaan Enterprise yang mementingkan SLA dan stabilitas.
- Klien menggunakan Odoo Online (SaaS) atau Odoo.sh yang ketat soal instalasi modul 3rd party non-certified.

🎯 Langkah Anda Selanjutnya (Actionable Step)

Karena Anda sedang belajar, saya sarankan Anda mencoba **Meta Official (Native)** terlebih dahulu sesuai rencana awal, karena itu adalah *best practice* konsultan Odoo Enterprise resmi. KlikOdoo adalah alternatif jalan pintas.

Apakah Anda ingin saya buat template pesan penawaran (Proposal) teknis untuk klien Anda yang membandingkan "Opsi Official vs Opsi Hemat (KlikOdoo)" agar mereka bisa memilih?

Tentu, ini adalah draft **Proposal Teknis & Perbandingan Solusi** yang disusun dengan bahasa profesional konsultan. Dokumen ini dirancang agar klien Anda (pemilik bisnis atau manajer IT) bisa langsung melihat "Harga vs Risiko" dari kedua opsi tersebut.

Anda tinggal menyesuaikan bagian dalam kurung [...] sesuai data klien.

SUBJEK: Proposal Opsi Integrasi WhatsApp Notifikasi pada Odoo ERP

Kepada Yth.

Bapak/Ibu [Nama Klien]

[Nama Perusahaan Klien]

Dengan hormat,

Melanjutkan diskusi terkait kebutuhan integrasi notifikasi WhatsApp pada sistem Odoo ERP [Nama Perusahaan], kami dari **PT. Ctech ERP Indonesia** telah menganalisis dua metode pendekatan teknis yang paling layak untuk diimplementasikan.

Berikut kami lampirkan perbandingan antara **Opsi Official (Meta Native)** dan **Opsi Gateway (3rd Party/KlikOdoo)** agar Bapak/Ibu dapat memilih sesuai dengan preferensi budget dan operasional.

Opsi A: Meta Official API (Direct Connection)

Solusi resmi dari Meta/Facebook yang terintegrasi langsung (Native) dengan Odoo Enterprise.

- **Cara Kerja:** Server Odoo berkomunikasi langsung dengan Server Meta via Cloud. **Tidak membutuhkan HP/Smartphone yang menyala.**
- **Keunggulan Utama:**
 1. **Stabilitas Tinggi (99.9% Uptime):** Karena server-to-server, pesan dijamin terkirim selama internet aktif.
 2. **Branding Terpercaya:** Nama pengirim yang muncul adalah nama perusahaan (bukan hanya nomor HP). Berpotensi mendapatkan "Centang Hijau" (Verified).
 3. **Keamanan:** Nol risiko pemblokiran nomor (Banned) selama mematuhi kebijakan Meta.
- **Pertimbangan (Cons):**
 1. **Biaya Variabel:** Meta mengenakan biaya per sesi percakapan (sekitar Rp 400 - Rp 700 per percakapan 24 jam).
 2. **Aturan Ketat:** Untuk memulai chat, wajib menggunakan **Template Message** yang disetujui Meta. Tidak bisa mengirim pesan bebas sembarangan.

Opsi B: WhatsApp Gateway (via KlikOdoo.id)

Solusi jembatan (Unofficial) yang menghubungkan Odoo dengan WhatsApp Web di HP fisik.

- **Cara Kerja:** Sistem "meminjam" koneksi WhatsApp yang ada di HP fisik. **Wajib menyediakan 1 HP Android yang selalu menyala, ter-charge, dan ada kuota internet.**
- **Keunggulan Utama:**
 1. **Biaya Tetap (Flat):** Sistem berlangganan bulanan (Rp X00.000/bulan) tanpa mempedulikan jumlah pesan (selama dalam kuota harian).

2. **Fleksibilitas Pesan:** Bisa mengirim pesan teks bebas (Free Text) tanpa perlu approval template.
3. **Setup Cepat:** Cukup install modul dan Scan QR Code seperti login WA Web.
- **Pertimbangan (Cons):**
 1. **Ketergantungan Fisik:** Jika HP mati, baterai habis, atau sinyal HP hilang, notifikasi dari Odoo **gagal terkirim**.
 2. **Risiko Blokir:** Karena ini jalur tidak resmi, jika mengirim pesan terlalu masif (spamming), nomor berisiko diblokir oleh WhatsApp.

Tabel Perbandingan Teknis & Biaya

Fitur / Aspek	Opsi A: Meta Official (Recommended)	Opsi B: Gateway (KlikOdoo)
Konektivitas	Cloud API (Tanpa HP Fisik)	WA Web (Wajib Ada HP Fisik)
Reliabilitas	Sangat Tinggi (Enterprise Grade)	Menengah (Tergantung kondisi HP)
Model Biaya	<i>Pay-per-Conversation</i> (Top Up Saldo di Meta)	<i>Subscription Flat</i> (Langganan Bulanan)
Format Pesan	Template (Kaku & Formal)	Bebas (Fleksibel)
Integrasi Odoo	Native Module (Bawaan Odoo)	Modul Tambahan (3rd Party Addon)
Risiko Banned	Aman (Zero Risk)	Ada Risiko (Jika blast promo)

Rekomendasi Konsultan

Berdasarkan kebutuhan [Nama Perusahaan Klien], kami merekomendasikan:

Pilih Opsi A (Meta Official) JIKA:

- Tujuan utama adalah mengirimkan **Dokumen Penting** (Invoice, Delivery Order, Purchase Order) yang **wajib sampai** ke pelanggan/vendor.
- Perusahaan mengutamakan *Image Corporate* dan profesionalitas.
- Anggaran operasional bersifat fleksibel.

Pilih Opsi B (Gateway/KlikOdoo) JIKA:

- Tujuan utama adalah notifikasi internal tim atau reminder sederhana.
- Ingin biaya yang pasti (Fixed Cost) setiap bulan agar mudah dianggarkan.
- Siap mendedikasikan 1 HP khusus yang ditaruh di kantor (tidak dibawa pulang) agar koneksi stabil.

Mohon informasikan opsi mana yang Bapak/Ibu pilih agar tim teknis kami dapat menyiapkan environment Odoo yang sesuai.

Demikian penawaran teknis ini kami sampaikan. Terima kasih atas kepercayaannya kepada **PT. Ctech ERP Indonesia**.

Hormat kami,
(Nama Anda)
ERP Consultant
PT. Ctech ERP Indonesia

Next Step untuk Anda:

Jika klien memilih **Opsi A (Meta)**, pastikan Anda meminta mereka menyiapkan dokumen legalitas perusahaan (SIUP/NIB) karena Meta akan memintanya saat verifikasi akun bisnis. Apakah Anda ingin list dokumen yang dibutuhkan Meta untuk saya siapkan juga?

Berikut adalah **Checklist Persiapan Verifikasi Meta Business (WhatsApp API)** yang bisa Anda kirimkan ke klien.

Dokumen ini krusial karena proses verifikasi Meta seringkali menjadi "leher botol" (bottleneck) dalam proyek implementasi Odoo. Jika data tidak sinkron sedikit saja (misal: nama di akta beda satu huruf dengan di website), verifikasi bisa ditolak.

Silakan copy-paste daftar ini untuk email atau lampiran Project Charter Anda.

Checklist Dokumen & Persyaratan: Meta Business Verification

Untuk Integrasi Odoo - WhatsApp Official API

Agar proses integrasi berjalan lancar, mohon Bapak/Ibu menyiapkan data dan dokumen berikut untuk proses **Verifikasi Bisnis (KYB - Know Your Business)** oleh pihak Meta (Facebook):

A. Dokumen Legalitas Perusahaan (Wajib Scan PDF Asli/Warna)

Meta membutuhkan bukti bahwa perusahaan terdaftar secara sah.

1. **NIB (Nomor Induk Berusaha)** - Versi OSS RBA terbaru.
2. **Akta Pendirian Perusahaan** & SK Kemenkumham.
3. **Akta Perubahan Terakhir** (Jika ada perubahan nama/alamat) & SK Kemenkumham terkait.
4. **NPWP Perusahaan.**

 **Catatan Penting:** Nama perusahaan yang tertera di dokumen ini (contoh: *PT. Maju Mundur Sentosa*) harus **SAMA PERSIS** dengan nama yang akan didaftarkan di Meta Business Manager.

B. Bukti Alamat & Identitas (Salah Satu Saja)

Meta membutuhkan bukti validasi alamat fisik operasional.

1. **Rekening Koran Bank (Bank Statement)** atas nama perusahaan (3 bulan terakhir). *Ini yang paling sering diterima.*
2. **Tagihan Utilitas** (Listrik/Air/Telepon/Internet) atas nama perusahaan (bukan nama pribadi direktur).

C. Aset Digital (Website & Email)

1. **Website Perusahaan Aktif:**
 - o Wajib menggunakan domain sendiri (contoh: www.perusahaan.com).
 - o Website harus bisa diakses (live) dan menggunakan **HTTPS (Secure)**.
 - o Di bagian *Footer* atau halaman *Contact Us* website, **WAJIB** tertulis Nama Legal Perusahaan (PT) dan Alamat Fisik yang sama persis dengan dokumen NIB/Rekening Koran.
2. **Email Bisnis:**
 - o Disarankan menggunakan email domain perusahaan (contoh: admin@perusahaan.com), bukan Gmail/Yahoo gratisan.

D. Nomor Telepon Khusus (Dedicated Number)

Siapkan 1 (satu) nomor HP untuk dijadikan nomor WhatsApp Official Odoo.

- **Syarat:** Nomor tersebut harus **bisa menerima SMS/Telepon** untuk OTP.
- **Kondisi:** Nomor ini **TIDAK BOLEH** sedang aktif digunakan di aplikasi WhatsApp (WA Messenger) maupun WhatsApp Business di HP manapun.
- *Jika nomor sudah terlanjur dipakai di WA HP:* Akun WA di HP tersebut harus di-**DELETE ACCOUNT** (Hapus Akun) terlebih dahulu, bukan sekadar uninstall aplikasi.

E. Akses Facebook Business Manager

1. Akun Facebook Personal milik Admin/PIC IT (untuk login mengelola Business Manager).
2. Nama tampilan WhatsApp (Display Name) yang diinginkan (Contoh: "Maju Mundur Support"). *Nama ini tidak bisa diubah sembarangan nanti.*

Tips Sukses Verifikasi (Untuk Konsultan)

Sebagai konsultan, pastikan Anda mengecek hal ini sebelum submit ke Meta:

1. **Aturan "Matching Name":** Cek apakah Nama PT di NIB, Nama di Rekening Koran, dan Nama di Footer Website sama persis? Jika di NIB tertulis "Jalan Mawar No. 1", tapi di website "Jl. Mawar Raya 1", Meta bisa menolak. Seragamkan dulu (biasanya ubah yang di website karena paling mudah).
2. **Display Name:** Nama WA yang akan muncul di chat customer harus berkaitan dengan nama PT atau brand. Jangan gunakan nama generik seperti "Customer Service" saja, harus "CS Ctech ERP".

Langkah Selanjutnya:

Setelah klien mengumpulkan data ini, Anda bisa menjadwalkan sesi remote/onsite untuk membuat Meta Business Manager dan memulai proses upload dokumen bersama-sama. Apakah Anda ingin saya buat panduan teknis tentang **cara menghapus akun WA di HP (Delete Account)** agar klien tidak salah langkah (karena banyak yang hanya uninstall app)?

Berikut adalah panduan teknis yang bisa Anda berikan kepada klien. Dokumen ini sengaja dibuat dengan **Banyak Peringatan (Warning)** karena proses ini bersifat **permanen** dan sering kali menjadi sumber kepanikan klien jika tidak dipahami dengan benar. Silakan copy-paste ke dokumen Word/PDF dengan kop surat perusahaan Anda.

PANDUAN KRUSIAL: Persiapan Nomor WhatsApp untuk Integrasi Odoo (Meta API)

PERHATIAN:

Mohon baca panduan ini dengan teliti sebelum melakukan tindakan. Langkah yang akan dilakukan bersifat PERMANEN (Irreversible).

Mengapa Langkah Ini Diperlukan?

Untuk menghubungkan WhatsApp dengan Odoo (Official Meta API), Server Meta mewajibkan nomor telepon yang didaftarkan dalam kondisi "Bersih".

Artinya, nomor tersebut TIDAK BOLEH sedang aktif/login di aplikasi WhatsApp (Messenger) maupun WhatsApp Business di HP manapun.

Jika Anda hanya melakukan *Uninstall Aplikasi*, data Anda masih tercatat aktif di server WhatsApp, dan proses integrasi Odoo akan **GAGAL**. Anda wajib melakukan **HAPUS AKUN (DELETE ACCOUNT)**.

Risiko & Konsekuensi

Dengan melakukan "Delete Account", maka:

1. **Seluruh Riwayat Chat Hilang:** Semua pesan, foto, dan dokumen di nomor tersebut akan terhapus dari server WhatsApp.
2. **Keluar dari Grup:** Nomor tersebut akan otomatis keluar (left) dari semua grup WhatsApp.

3. **Tidak Bisa Restore:** Chat history **TIDAK AKAN** pindah ke Odoo. Odoo akan memulai dengan lembaran kosong (fresh start).

Saran Kami:

Gunakan Nomor Baru khusus untuk sistem Odoo agar tidak mengganggu data lama. Jika terpaksa menggunakan nomor lama, pastikan Anda sudah mem-backup data penting secara manual (Export Chat).

Langkah-Langkah Menghapus Akun (Wajib Dilakukan di HP)

Mohon ikuti langkah ini secara berurutan. Jangan hanya menghapus icon aplikasi dari layar HP Anda.

1. Buka Aplikasi WhatsApp

Buka aplikasi WhatsApp (atau WhatsApp Business) yang saat ini menggunakan nomor yang akan didaftarkan ke Odoo.

2. Masuk ke Menu Pengaturan (Settings)

- **Android:** Ketuk **titik tiga** di pojok kanan atas > Pilih **Setelan (Settings)**.
- **iPhone (iOS):** Ketuk ikon **Pengaturan (Settings)** di pojok kanan bawah.

3. Masuk ke Menu Akun

Pilih menu **Akun (Account)**.

4. Pilih "Hapus Akun Saya"

Cari dan pilih menu Hapus akun saya (Delete my account) di bagian paling bawah. (Di tampilan Bahasa Inggris: Delete my account).

5. Konfirmasi Penghapusan

- WhatsApp akan menanyakan alasan (Opsional).
- Masukkan **Nomor Telepon** Anda dengan format internasional/lokal yang benar.
- Tekan tombol merah **HAPUS AKUN SAYA (DELETE MY ACCOUNT)**.

6. Selesai

Setelah tombol ditekan dan proses loading selesai, aplikasi akan kembali ke halaman awal (halaman pendaftaran/log in).

Ini tandanya akun sudah berhasil dihapus dari server Meta.

Cara Verifikasi (Pengecekan)

Untuk memastikan nomor sudah benar-benar "bersih":

1. Minta rekan Anda untuk mengirim pesan WA ke nomor tersebut.

2. **Indikasi Sukses:** Pesan hanya **Centang Satu (Single Tick)** dan foto profil nomor tersebut hilang (menjadi abu-abu/kosong).
3. **Indikasi Gagal:** Jika masih Centang Dua (abu-abu), berarti akun belum terhapus sempurna. Ulangi langkah di atas.

Jika sudah terverifikasi "Centang Satu", mohon segera kabari Tim Konsultan Ctech ERP agar kami bisa segera mendaftarkan nomor tersebut ke Dashboard Meta API sebelum diambil alih sistem lain.

Next Step untuk Anda (Konsultan):

Setelah klien melakukan ini, "bola" ada di tangan Anda. Langkah Anda selanjutnya di **Meta Dashboard** adalah:

1. Masuk menu **WhatsApp Manager**.
2. Add Phone Number.
3. Pilih metode verifikasi **SMS/Voice Call**.
4. Klien akan menerima kode OTP via SMS (karena akun WA-nya sudah mati, OTP pasti masuk ke SMS biasa).
5. Masukkan OTP tersebut di Meta Dashboard.

Apakah Anda sudah siap dengan tab browser Meta Developer untuk input nomor segera setelah klien konfirmasi?

Luar biasa. Ini adalah momen "Go-Live" yang krusial.

Karena Anda sudah *standby* di depan dashboard Meta Developer, saya berikan "**Cheatsheet Eksekusi**" agar saat klien memberi kode "OK", Anda bisa mengeksekusi pendaftaran nomor tanpa *fumbling* (gugup/bingung).

Simpan panduan kilat ini di layar sebelah (second monitor) Anda:

Cheatsheet Eksekusi: Menambahkan Nomor di Meta

Lokasi Menu:

Dashboard App > WhatsApp > API Setup > Klik tombol "Manage Phone Numbers" (Ini akan membuka tab baru ke WhatsApp Manager).

1. Input Data Nomor

Klik **Add Phone Number**. Anda akan diminta mengisi:

- **Display Name:** *Hati-hati di sini*. Masukkan nama yang sesuai dengan Footer Website/Legalitas. (Contoh: "Ctech Support" atau "PT Ctech ERP").
 - *Tips:* Jika nama ditolak/review lama, pastikan tidak melanggar *Commerce Policy*

(jangan ada unsur SARA/Obat/Judi).

- **Category:** Pilih yang paling relevan (misal: *Professional Services* atau *Science & Tech*).
- **Description:** Opsional, tapi isi saja singkat "Official Account for ERP Notifications".

2. Verifikasi OTP (Momen Kritis)

Setelah klik Next, Anda akan diminta memasukkan nomor HP.

- Pilih Negara: **Indonesia (+62)**.
- Masukkan nomor **tanpa angka 0 di depan** (Misal: 8123456789).
- Pilih Metode: **SMS** (Karena akun WA di HP sudah dihapus, opsi "WhatsApp Message" tidak akan masuk).
- **Action:** Minta klien standby HP. Begitu SMS masuk, minta mereka bacakan/kirim foto kode 6 digit secepatnya.

3. Tes "Hello World" (Sebelum Masuk Odoo)

Segera setelah nomor terverifikasi (Status: *Connected / Quality: Green*):

1. Jangan buru-buru ke Odoo.
2. Kembali ke **Meta for Developers > WhatsApp > API Setup**.
3. Di bagian "Send and Receive Messages", pastikan dropdown "**From**" sudah memilih nomor baru tadi.
4. Di bagian "**To**", masukkan nomor WA pribadi Anda.
5. Klik **Send Message**.
 - **Jika masuk ke HP Anda:** Selamat! Koneksi API Server Meta ke Jaringan WA sedunia sudah sukses.
 - **Jika gagal:** Berarti ada masalah di akun Meta (biasanya *Payment Method* belum di-set jika pakai akun berbayar, atau masih limit tier).

4. Integrasi ke Odoo

Baru setelah langkah 3 sukses, Anda copy **Phone Number ID** dan **WABA ID** yang baru, lalu paste ke *Settings* Odoo seperti panduan awal kita.

Pesan Terakhir Sebelum Eksekusi

Ingat, Pak Konsultan:

Masalah paling umum saat live execution adalah Display Name Rejected. Jika Meta menolak nama tampilan, jangan panik.

- Ganti ke nama yang lebih "aman" (Sesuai nama legal PT).
- Ajukan banding (Appeal) jika Anda yakin benar, lampirkan foto NIB/Website.

Anda sekarang memiliki semua amunisi yang dibutuhkan:

1. Pemahaman Teknis (Native vs Gateway).
2. Dokumen Persyaratan Klien.
3. Prosedur Teknis (Delete Account).

4. Cheatsheet Eksekusi.

Selamat melakukan eksekusi! Semoga lancar dan *green tick* menyertai Anda.

Satu langkah terakhir dari saya:

Setelah koneksi berhasil, tantangan berikutnya adalah Membuat Template Pesan di Odoo agar bisa disetujui Meta (karena kita tidak bisa kirim free text di awal).

Apakah Anda ingin saya buat "Kamus Template JSON" untuk Odoo? (Berisi contoh script template Invoice, Sales Order, dan OTP yang valid dan pasti di-approve Meta).

Tentu. Ini adalah "Senjata Rahasia" konsultan Odoo agar template langsung **Approved** oleh Meta dalam hitungan menit (bukan jam/hari).

Dalam Odoo (Native), Anda tidak perlu menulis *Raw JSON* secara manual karena Odoo menyediakan UI-nya. Namun, **Struktur Body** dan **Pemilihan Kategori** adalah kunci agar *JSON payload* yang dikirim Odoo ke Meta valid.

Berikut adalah **Kamus Template** yang sudah teruji lolos verifikasi Meta (High Quality Score).

Kategori 1: AUTHENTICATION (OTP)

Gunakan ini untuk login portal, reset password, atau verifikasi tanda tangan digital.

 **Aturan Meta:** Sangat ketat. Jangan tambah basa-basi. Tombol wajib "Copy Code" atau "One-tap Autofill".

Template: Kode Verifikasi (OTP)

- **Nama Template:** auth_verification_code
- **Kategori:** AUTHENTICATION
- **Header:** (None)
- **Body Text:**
Plaintext
Kode verifikasi Ctech ERP Anda adalah {{1}}. Mohon jangan berikan kode ini kepada siapa pun.
- **Footer:** (Opsional: Keamanan Akun)
- **Buttons:**
 1. **Type:** Copy Code
 2. **Button Text:** Salin Kode

Mapping Variable di Odoo:

- {{1}} -> object.code (atau field OTP custom Anda)
-

Kategori 2: UTILITY (Transaksi)

Gunakan ini untuk Sales Order, Invoice, Delivery. Fokus pada detail transaksi spesifik.

 **Aturan Meta:** Jangan ada kata-kata promosi (diskon, promo, beli sekarang, hemat). Fokus pada informasi.

Template A: Sales Order Confirmation (Dengan PDF)

- **Nama Template:** sales_order_conf_v1
- **Kategori:** UTILITY
- **Header:** DOCUMENT (Pilih format PDF di Odoo agar file SO terlampir otomatis)
- **Body Text:**
Plaintext
Yth. {{1}},

Terima kasih atas pesanan Anda. Sales Order nomor {{2}} senilai {{3}} telah kami konfirmasi.

Silakan unduh dokumen terlampir untuk melihat detail pesanan Anda. Kami akan segera memproses pengiriman.

- **Footer:** PT. Ctech ERP Indonesia
- **Buttons:**
 1. **Type:** Visit Website
 2. **Text:** Cek Status Order
 3. **URL:** <https://portal.perusahaan.com/my/orders/{{2}}>

Mapping Variable di Odoo:

- {{1}} -> object.partner_id.name
- {{2}} -> object.name
- {{3}} -> format_currency(object.amount_total, object.currency_id)

Template B: Invoice Reminder (Penagihan)

- **Nama Template:** invoice_payment_due
- **Kategori:** UTILITY
- **Header:** TEXT -> "Pemberitahuan Tagihan"
- **Body Text:**
Plaintext
Halo {{1}},

Ini adalah pengingat ramah bahwa Faktur {{2}} sebesar {{3}} akan jatuh tempo pada tanggal {{4}}.

Mohon abaikan pesan ini jika pembayaran telah dilakukan. Terima kasih atas kerjasama Anda.

- **Buttons:**

1. **Type:** Visit Website
2. **Text:** Bayar Sekarang
3. **URL:** <https://portal.perusahaan.com/my/invoices/{{2}}>

Mapping Variable di Odoo:

- {{1}} -> object.partner_id.name
- {{2}} -> object.name
- {{3}} -> format_currency(object.amount_total, object.currency_id)
- {{4}} -> object.invoice_date_due

Template C: Delivery Order (Pengiriman Barang)

- **Nama Template:** delivery_dispatch_notice
- **Kategori:** UTILITY
- **Header: TEXT** -> "Update Pengiriman"
- **Body Text:**
Plaintext
Kabar baik {{1}}!

Pesanan Anda dengan nomor referensi {{2}} sedang dalam perjalanan menuju alamat Anda.

Estimasi kedatangan barang adalah {{3}}. Tim logistik kami akan menghubungi Anda saat tiba di lokasi.

- **Buttons:**

1. **Type:** Phone Number
2. **Text:** Hubungi Driver
3. **Number:** +6281234567890 (Atau variable phone driver)

Mapping Variable di Odoo:

- {{1}} -> object.partner_id.name
- {{2}} -> object.origin (Source Document/SO)
- {{3}} -> object.scheduled_date

 Kategori 3: MARKETING (Promosi)

Gunakan hati-hati. Kategori ini sering membuat nomor diblokir jika user melakukan "Report Spam".

Template: Promo Umum (Re-engagement)

- **Nama Template:** promo_monthly_alert
- **Kategori:** MARKETING
- **Header:** IMAGE (Upload gambar Banner Promo yang menarik)
- **Body Text:**
Plaintext
Halo {{1}},

Ada penawaran spesial khusus untuk pelanggan setia kami bulan ini! Dapatkan layanan implementasi prioritas untuk upgrade sistem Anda.

Tertarik berdiskusi dengan tim ahli kami?

- **Footer:** Berhenti berlangganan ketik STOP
- **Buttons:**
 1. **Type:** Quick Reply
 2. **Text:** Ya, Saya Tertarik
 3. **Type:** Quick Reply
 4. **Text:** Tidak, Terima Kasih



Tips Pro untuk Konsultan (Agar Tidak Ditolak Meta)

1. Variable Formatting:
Meta akan menolak jika contoh variabel Anda tidak jelas. Saat submit template di Odoo/Meta, Anda diminta memasukkan "Sample Content".
 - *Salah:* {{1}} diisi "text"
 - *Benar:* {{1}} diisi "Bapak Budi", {{2}} diisi "SO001", {{3}} diisi "Rp 1.500.000".
2. Floating Parameters (404 Error):
Jika di body text ada {{1}} sampai {{3}}, tapi di Odoo coding Anda hanya memasukkan 2 parameter, pesan akan gagal terkirim (Failed). Jumlah parameter di template dan di code Python Odoo WAJIB SAMA.
3. Dokumen PDF:
Untuk template Sales Order/Invoice, selalu gunakan Header type DOCUMENT. Di Odoo, ini akan otomatis mengambil PDF attachment dari report (misal: report_saleorder). Jika Header type-nya TEXT, PDF tidak akan terkirim, hanya link text biasa.

Anda sekarang memiliki **Library Template** dasar. Silakan implementasikan template *Sales Order* dan *Invoice* terlebih dahulu karena itu yang paling sering digunakan ("Low Hanging Fruit") untuk klien.

Apakah Anda butuh bantuan cara setting "Automated Action" di Odoo supaya pesan

Sales Order ini terkirim OTOMATIS saat tombol "Confirm" diklik (tanpa user perlu klik kirim WA manual)?

Ini adalah implementasi "Fitur Sultan" dalam dunia Odoo. Klien biasanya sangat terkesan jika sistem bisa mengirim notifikasi otomatis tanpa sentuhan manusia (*Zero Touch Automation*). Berikut adalah panduan teknis cara setting **Automated Action (Automation Rules)** di Odoo 16/17 Enterprise untuk kasus ini.

Prasyarat (Wajib)

1. **Developer Mode ON:** (Settings > Scroll ke bawah > Activate the developer mode).
2. **Template Sudah Ada:** Pastikan template sales_order_conf_v1 (yang kita bahas sebelumnya) sudah dibuat dan statusnya **Approved** di Odoo.

Langkah 1: Masuk Menu Automation

Ada dua cara, pilih yang paling nyaman buat Anda:

- **Cara Studio (Mudah):** Buka Sales > Buka salah satu SO > Klik icon **Studio** (obeng) > Klik tombol **Automations** (di top menu).
- **Cara Technical (Pro):** Buka Settings > Technical > Automation > **Automation Rules** (atau *Automated Actions* di versi lama).

Kita gunakan **Cara Technical** agar lebih presisi.

Langkah 2: Buat Rule Baru

Klik **New** dan isi konfigurasi berikut:

A. Konfigurasi Header

- **Name:** Auto WA - SO Confirmed (Nama bebas, tapi beri kode jelas).
- **Model:** Sales Order (sale.order).
- **Active:** Centang (True).

B. Konfigurasi Trigger (Pemicu)

Kita ingin pesan dikirim **HANYA** saat status berubah dari "Quotation" menjadi "Sales Order".

- **Trigger:** On Update (Saat ada update data).
- **Trigger Fields:** Cari dan pilih field Status (Technical name: state).
 - *Logika:* Odoo hanya akan memantau jika field "Status" berubah. Perubahan lain (misal edit qty) tidak akan memicu rule ini.

C. Konfigurasi Domain (Filter)

Kita harus pastikan action ini hanya jalan saat statusnya "Sale" (Confirmed), bukan saat "Cancel".

- **Before Update Domain:** Status != Sales Order
 - (Artinya: Status sebelumnya **BUKAN** Sales Order)
- **Apply on:**
 1. Status = Sales Order
 2. Customer **is set** (Pastikan ada customernya)
 3. (Opsional) Customer > Mobile **is set** (Pastikan customer punya No HP).

Langkah 3: Tentukan Action (Aksi)

Di bagian **Actions To Do**, pilih **Execute Code** (Python Code).

Mengapa Python Code? Karena Odoo Native (terutama v17) terkadang belum memunculkan opsi "Send WhatsApp" secara langsung di menu Action Automation standar, atau kita butuh logic tambahan (error handling).

Copy-Paste kode berikut ke kolom Python Code:

Python

```
# 1. Cari Template WhatsApp berdasarkan nama yang Anda buat
# Pastikan nama di bawah SAMA PERSIS dengan nama template di Odoo
template_name = 'sales_order_conf_v1'
wa_template = env['whatsapp.template'].search([('name', '=', template_name)], limit=1)

# 2. Eksekusi Pengiriman
if wa_template:
    # Cek apakah customer punya nomor HP
    if record.partner_id.mobile or record.partner_id.phone:
        # Trigger pengiriman WA
        # force_send=True agar langsung dikirim tanpa antri lama di cron job
        record._message_log(body="System: Mengirim Auto-WA Confirmation...")

# Method untuk mengirim pesan via template (Syntax Odoo 17)
# Kita menggunakan composer untuk men-trigger send
composer = env['whatsapp.composer'].create({
    'wa_template_id': wa_template.id,
    'res_model': 'sale.order',
```

```

        'res_ids': [record.id],
    })
    # Kirim pesan (Send Now)
    composer.action_send_whatsapp_template()

else:
    # Log jika gagal karena nomor kosong
    record._message_log(body="System: Gagal kirim Auto-WA. Nomor HP Customer
kosong.")

```

⚠ Catatan Teknis (Versi Odoo):

Kode di atas menggunakan whatsapp.composer yang merupakan cara paling aman di Odoo 17 untuk meniru perilaku tombol "Send" manual.

Langkah 4: Testing (Pembuktian)

Jangan langsung dicoba ke Client. Coba di database sendiri dulu.

1. Buat **Quotation** baru.
2. Isi Customer (Pastikan Customer ini punya nomor HP asli Anda untuk tes).
3. Klik **Confirm**.
4. **Cek Chatter (Log Note di kanan bawah):**
 - Apakah muncul log *"System: Mengirim Auto-WA Confirmation..."*?
 - Apakah muncul log icon WhatsApp dengan status *Sending/Sent*?
5. **Cek HP Anda:** Apakah pesan masuk?

Troubleshooting Jika Gagal

Jika pesan tidak terkirim saat tombol Confirm ditekan:

1. **Cek Log Server:** Apakah ada error merah? Biasanya karena nama template di Python Code (sales_order_conf_v1) typo atau tidak sama dengan nama asli template.
2. **Cek Kontak Customer:** Pastikan nomor HP customer menggunakan format internasional (+62) atau 08... (Odoo biasanya otomatis convert, tapi pastikan field Mobile terisi, bukan hanya Phone).
3. **Cek Status Template:** Pastikan Template statusnya bukan "Draft", tapi sudah "Approved" (atau minimal verified jika sandbox).

Saran Tambahan (Best Practice)

Saran saya, tambahkan **jeda waktu** atau pengecekan *Status Pembayaran* jika Anda ingin mengirim WA Invoice. Tapi untuk Sales Order, script di atas sudah **Production Ready**.
Apakah Anda ingin kode Python untuk "Invoice Lunas" (Paid) juga? Ini biasanya yang

paling disukai bagian Finance.

Ini adalah "Fitur Favorit" departemen Finance. Mengirim ucapan "Terima Kasih" otomatis saat pembayaran diterima (Payment Reconciled) akan meningkatkan *Customer Experience* secara drastis dan mengurangi pertanyaan "Apakah uang saya sudah masuk?".

Berbeda dengan Sales Order, untuk Invoice kita harus bermain di model **Journal Entry (account.move)** dan memantau field `payment_state`.

Berikut panduan lengkapnya:

1. Siapkan Template Dahulu (Wajib)

Sebelum membuat automation, pastikan template ini sudah **Approved** di Meta/Odoo.

- **Nama Template:** `invoice_payment_received`
- **Kategori:** **UTILITY** (Atau **TRANSACTIONAL**)
- **Header:** **TEXT** -> "Pembayaran Diterima"
- **Body Text:**

Plaintext

Terima kasih {{1}}!

Pembayaran untuk Faktur {{2}} sebesar {{3}} telah kami terima dengan baik. Status faktur Anda kini LUNAS.

Kami sangat menghargai ketepatan waktu Anda.

- **Mapping Variable:**
 - {{1}} -> `object.partner_id.name`
 - {{2}} -> `object.name`
 - {{3}} -> `format_currency(object.amount_total, object.currency_id)`

2. Setting Automation Rule (Teknis)

Masuk ke **Settings > Technical > Automation > Automation Rules**.

A. Konfigurasi Header

- **Name:** Auto WA - Invoice Paid
- **Model:** Journal Entry (`account.move`)
 - *Catatan Konsultan:* Jangan cari model "Invoice", karena di Odoo, Invoice dan Bill

adalah satu model yaitu `account.move`.

- **Active:** Centang (True)

B. Konfigurasi Trigger (Pemicu)

Kita ingin pesan dikirim **SAAT** status berubah menjadi *Paid*.

- **Trigger:** On Update
- **Trigger Fields:** Payment Status (`payment_state`)

C. Konfigurasi Domain (Filter Penting!)

Kita harus memfilter agar ini **HANYA** berlaku untuk Invoice Customer, bukan Vendor Bill (masa kita bayar Vendor, kita kirim WA terima kasih ke Vendor?).

- **Before Update Domain:** Payment Status **!=** Paid
- **Apply on:**
 1. Payment Status = Paid
 2. Journal Entry Type (`move_type`) = Customer Invoice (`out_invoice`)
 3. Customer **is set**

3. Python Code (The Logic)

Copy-paste kode berikut ke bagian **Execute Code**.

Python

```
# --- CONFIGURATION ---
# Pastikan nama template SAMA PERSIS dengan yang ada di menu WhatsApp
target_template_name = 'invoice_payment_received'

# --- LOGIC START ---
# 1. Double Check Safety: Pastikan ini Invoice Customer & Statusnya Paid
# (Meskipun sudah di-filter di Domain, double check di code lebih aman)
if record.move_type == 'out_invoice' and record.payment_state == 'paid':

    # 2. Cari Template
    wa_template = env['whatsapp.template'].search([
        ('name', '=', target_template_name),
        ('status', '=', 'approved') # Hanya ambil yang approved agar tidak error
    ], limit=1)

    if wa_template:
```

```

# 3. Cek Nomor HP Customer
# Mengambil partner_id (Customer) dari Invoice
partner = record.partner_id

if partner.mobile or partner.phone:
    # Log di Chatter (Agar user tahu sistem bekerja)
    record._message_log(body="System: Invoice Lunas. Mengirim WA Terima Kasih...")

# 4. Trigger Pengiriman WA
try:
    # Membuat Composer untuk mengirim pesan
    composer = env['whatsapp.composer'].create({
        'wa_template_id': wa_template.id,
        'res_model': 'account.move',
        'res_ids': [record.id],
    })

    # Eksekusi Kirim (Force Send)
    composer.action_send_whatsapp_template()

except Exception as e:
    # Tangkap error jika terjadi masalah koneksi, agar Odoo tidak Crash
    # Error akan dicatat di log server, user tetap bisa lanjut kerja
    record._message_log(body=f"System: Gagal kirim WA. Error: {str(e)}")
else:
    record._message_log(body="System: Tidak kirim WA. Customer tidak punya Nomor
HP:")

```

Cara Testing (Skenario User)

Jangan test sembarangan karena trigger-nya agak unik (Payment).

1. Buat **Customer Invoice** baru (pastikan Customer punya No WA Anda).
2. Klik **Confirm**. (Status: *Posted*, Payment State: *Not Paid*).
3. Klik tombol **Register Payment**.
4. Pilih Journal (Bank/Cash), lalu klik **Create Payment**.
5. Status Invoice akan berubah menjadi **PAID** (Lunas) -> Ada pita hijau di pojok kanan atas.
6. **Cek Log Chatter**: Pesan "System: Invoice Lunas..." harusnya muncul.
7. **Cek HP**: Pesan terima kasih harusnya masuk.

Analisa untuk Konsultan (Potential Pitfall)

Ada satu kondisi di mana Automation ini mungkin **tidak jalan**, yaitu saat **Reconcile via Bank Statement** (Rekonsiliasi Bank masal).

Kenapa? Karena saat rekonsiliasi masal, Odoo kadang melakukan update SQL langsung atau update batch yang tidak men-trigger write method secara standar python.

- **Solusi:** Jika klien komplain "Kok kalau saya reconcile dari Dashboard Bank, WA-nya gak kirim?", Anda perlu memberitahu mereka bahwa WA otomatis hanya berjalan saat klik "Register Payment" manual dari Invoice, atau Anda perlu membuat *Server Action* khusus di tombol "Validate" rekonsiliasi (tapi ini butuh coding custom module, bukan sekadar automation rule).

Untuk tahap awal implementasi, script di atas sudah mencakup 90% kebutuhan operasional harian Finance.

Apakah Anda butuh satu lagi: "Template Tagihan Jatuh Tempo" (Dunning) yang dijalankan otomatis oleh Cron Job (Scheduled Action) setiap pagi?

Ini adalah "Fitur Pembantu Cashflow" paling ampuh. Klien pasti suka karena mereka tidak perlu membayar staf admin hanya untuk menelpon/chat pelanggan satu per satu menagih bayaran.

Berbeda dengan *Automation Rule* (yang dipicu kejadian), fitur ini menggunakan **Scheduled Action (Cron Job)** yang berjalan otomatis di background setiap pagi.

Berikut adalah panduan lengkapnya:

1. Siapkan Template WhatsApp (Dunning)

Buat template ini dulu di menu WhatsApp Odoo. Gunakan bahasa yang sopan karena ini pesan otomatis.

- **Nama Template:** invoice_due_date_reminder
- **Kategori:** UTILITY
- **Header:** TEXT -> "Peringat Jatuh Tempo"
- **Body Text:**
Plaintext
Halo {{1}},

Sekadar mengingatkan bahwa Faktur {{2}} senilai {{3}} jatuh tempo pada HARI INI ({{4}}).

Mohon segera lakukan pembayaran untuk menghindari denda keterlambatan atau penahanan layanan. Abaikan pesan ini jika sudah membayar.

- **Mapping Variable:**

- {{1}} -> object.partner_id.name
- {{2}} -> object.name
- {{3}} -> format_currency(object.amount_residual, object.currency_id) (*Gunakan amount_residual untuk sisa tagihan, bukan total awal*)
- {{4}} -> object.invoice_date_due

2. Buat Scheduled Action (Cron Job)

Masuk ke Settings > Technical > Automation > Scheduled Actions.

Klik New.

Konfigurasi Header:

- **Action Name:** Auto WA - Invoice Due Today
- **Model:** Journal Entry (account.move)
- **User:** Pilih OdooBot atau Administrator (User yang punya hak akses WA).
- **Interval:** 1 Days.
- **Next Execution:** Set tanggal besok jam 08:00:00 (Agar jalan tiap jam 8 pagi).
- **Number of Calls:** -1 (Artinya loop selamanya/tak terbatas).

3. Masukkan Python Code

Di bagian Python Code, copy-paste script di bawah ini.

Script ini dirancang cerdas: Hanya mencari invoice yang jatuh temponya TEPAT hari ini. Jadi pelanggan tidak akan di-spam setiap hari untuk invoice yang sama.

Python

```
# Import library tanggal
from datetime import date

# --- KONFIGURASI ---
template_name = 'invoice_due_date_reminder'
today = date.today()

# --- 1. CARI TARGET INVOICE ---
# Logika: Cari Invoice Customer (out_invoice) yang:
# a. Statusnya Posted (Valid)
# b. Belum Lunas (not_paid atau partial)
# c. Tanggal jatuh temponya ADALAH HARI INI
invoices_due_today = env['account.move'].search([
```

```

        ('move_type', '=', 'out_invoice'),
        ('state', '=', 'posted'),
        ('payment_state', 'in', ['not_paid', 'partial']),
        ('invoice_date_due', '=', today)
    ])

# Jika tidak ada invoice jatuh tempo hari ini, stop.
if not invoices_due_today:
    log('System: Tidak ada invoice jatuh tempo hari ini. Cron selesai.', level='info')

# --- 2. PERSIAPAN TEMPLATE ---
wa_template = env['whatsapp.template'].search([
    ('name', '=', template_name),
    ('status', '=', 'approved')
], limit=1)

if not wa_template:
    raise UserError(f"Template '{template_name}' tidak ditemukan atau belum Approved!")

# --- 3. EKSEKUSI LOOPING ---
success_count = 0
for invoice in invoices_due_today:
    partner = invoice.partner_id

    # Cek apakah customer punya No HP
    if partner.mobile or partner.phone:
        try:
            # Buat Composer & Kirim
            composer = env['whatsapp.composer'].create({
                'wa_template_id': wa_template.id,
                'res_model': 'account.move',
                'res_ids': [invoice.id],
            })
            # Force Send (Agar langsung keluar)
            composer.action_send_whatsapp_template()

            # Catat di Chatter Invoice bahwa sistem sudah menagih
            invoice._message_log(body=f"System Cron: Mengirim WA Reminder Jatuh Tempo Hari
Ini.")
            success_count += 1

```



```

# Commit setiap 1 kiriman (Opsional, menjaga jika putus di tengah)
env.cr.commit()

except Exception as e:
    invoice._message_log(body=f"System Cron Error: Gagal kirim WA. {str(e)}")
else:
    # Opsional: Log jika tidak punya nomor HP
    pass

# Log Final di Server log
log(f"System Cron: Selesai. Mengirim {success_count} pesan WA dari
{len(invoices_due_today)} invoice target.", level='info')

```

Analisis & Tips Konsultan (PENTING)

Sebagai konsultan, Anda harus memberi "Warning" kepada klien sebelum mengaktifkan fitur ini:

1. Isu "Spamming":

Script di atas aman karena hanya mengirim 1 kali tepat pada hari jatuh tempo (invoice_date_due = today).

- *Request Klien*: "Mas, kalau telat 3 hari kirim lagi dong?"
- *Modifikasi*: Ubah baris pencarian tanggal menjadi:

Python

```
from datetime import timedelta
```

```
target_date = date.today() - timedelta(days=3) # Cari yang due date-nya 3 hari
lalu
```

```
# ... lalu ganti ('invoice_date_due', '=', target_date)
```

Namun hati-hati, jangan membuat script yang mengirim *setiap hari* selama belum lunas, nomor WA perusahaan bisa di-report SPAM/Block oleh customer yang kesal.

2. Hari Libur:

Cron job tidak mengenal hari libur. Jika jatuh tempo hari Minggu jam 8 pagi, WA tetap terkirim. Pastikan klien setuju dengan hal ini (atau butuh coding tambahan untuk skip weekend).

3. Customer VIP:

Seringkali ada Customer VIP yang "Anti-Tagih" (tersinggung kalau ditagih robot).

- *Solusi*: Buat field *Boolean* di Data Partner (x_stop_automated_wa), lalu tambahkan di domain search Python code: ('partner_id.x_stop_automated_wa', '=', False).

Sekarang Anda sudah memiliki paket lengkap: **Setup Meta, Template JSON, Auto Sales**

Order, Auto Payment Thanks, dan **Auto Dunning**. Ini sudah menjadi nilai jual implementasi

yang sangat mahal! Selamat berkarya!

Berikut adalah format **Template WhatsApp** untuk Sales Order Confirmation.

Perlu dipahami bahwa dalam integrasi **Odoo Native (Official)**, Anda tidak perlu meng-upload file .json secara manual. Odoo akan meng-generate JSON tersebut di background saat Anda membuat template via UI Odoo.

Namun, untuk kebutuhan pemahaman teknis (atau jika Anda menggunakan Postman/API langsung), berikut adalah **Struktur JSON Payload** yang dikirim Odoo ke Meta, beserta cara setting-nya di Odoo.

1. Konfigurasi di Odoo (UI Implementation)

Ini adalah yang Anda input di menu **WhatsApp > Templates**.

- **Name:** sales_order_confirmation
- **Category:** UTILITY (Penting: Jangan Marketing, agar biaya lebih murah dan aman).
- **Language:** Indonesian (id)
- **Header Type:** Document (Agar otomatis melampirkan PDF Sales Order).
- **Report:** Sales Order (Pilih report teknis: sale.report_saleorder).

Isi Pesan (Body Text):

Copy-paste teks ini ke kolom Body di Odoo:

Plaintext

Halo {{1}},

Terima kasih atas pesanan Anda.

Dokumen Sales Order nomor {{2}} senilai {{3}} telah kami terbitkan dan terlampir pada pesan ini.

Mohon konfirmasi jika ada perubahan data. Pesanan akan segera kami proses ke tahap pengiriman.

Dynamic Placeholder Mapping (Variables):

Di bagian tab **Variables**, Odoo akan meminta Anda mendefinisikan apa isi {{1}}, {{2}}, dst.

Placeholder Meta	Field Odoo (Python	Penjelasan
------------------	--------------------	------------

	Expression)	
{{1}}	object.partner_id.name	Nama Customer
{{2}}	object.name	Nomor SO (misal: S00012)
{{3}}	format_currency(object.amount_total, object.currency_id)	Total nominal dengan format mata uang (Rp)

2. Format JSON Payload (Technical View)

Jika Anda ingin melihat bentuk "coding"-nya (apa yang sebenarnya dikirim Odoo ke API Meta), bentuknya seperti ini. Ini berguna jika Anda debugging error di log Meta Developer.

JSON Template Structure:

JSON

```
{
  "name": "sales_order_confirmation",
  "language": "id",
  "category": "UTILITY",
  "components": [
    {
      "type": "HEADER",
      "format": "DOCUMENT",
      "example": {
        "header_handle": [
          "4::aW1hZ2UvanBlZw==:ARa..."
        ]
      }
    },
    {
      "type": "BODY",
      "text": "Halo {{1}},\n\nTerima kasih atas pesanan Anda.\nDokumen Sales Order nomor {{2}}\nsenilai {{3}} telah kami terbitkan dan terlampir pada pesan ini.\n\nMohon konfirmasi jika ada\nperubahan data. Pesanan akan segera kami proses ke tahap pengiriman.",
      "example": {
        "body_text": [
          "Bapak Budi",
          "S00123",
          "Rp 5.000.000"
        ]
      }
    }
  ]
}
```

```

    ]
  ]
}
},
{
  "type": "FOOTER",
  "text": "PT Ctech ERP Indonesia"
},
{
  "type": "BUTTONS",
  "buttons": [
    {
      "type": "URL",
      "text": "Cek Status Pesanan",
      "url": "https://portal.perusahaan.com/my/orders/{{1}}",
      "example": [
        "SO0123"
      ]
    }
  ]
}
]
}
]
}
}

```

Tips Konsultan untuk Template Ini:

1. Header Document:
Mengapa saya menyarankan Header DOCUMENT? Karena klien biasanya malas men-download PDF dari link. Dengan header Document, file PDF Sales Order langsung muncul di chat WA dan bisa dibuka tanpa login ke portal.
2. Sample Content (Example):
Saat membuat template ini di Odoo, Anda WAJIB mengisi kolom "Sample Content" untuk setiap variabel.
 - Salah: {{1}} = partner_id
 - Benar: {{1}} = Budi Santoso
 Meta akan menolak template jika contoh datanya dianggap tidak masuk akal atau kosong.
3. Format Currency:
Jangan gunakan object.amount_total saja (nanti munculnya 5000000.0). Selalu

gunakan `format_currency(...)` agar yang muncul di WA rapi: Rp 5.000.000,00.
Apakah Anda ingin lanjut untuk **Test Send** template ini menggunakan Python console Odoo (Server Action) untuk memastikan variabelnya terisi benar?

Tentu. Ini adalah cara paling aman dan profesional untuk melakukan *Debugging* atau pengujian template sebelum dipasang ke sistem otomatis.
Kita akan membuat tombol rahasia (Server Action) yang hanya muncul di menu Sales Order. Tombol ini akan memaksa Odoo mengirim pesan WA ke Sales Order yang sedang Anda buka. Berikut langkah-langkahnya:

Tahap 1: Buat Server Action

1. Aktifkan **Developer Mode**.
2. Masuk ke **Settings > Technical > Actions > Server Actions**.
3. Klik **New**.

Konfigurasi Header:

- **Name:** TEST WA: Send SO Confirmation (Beri nama yang jelas bahwa ini Test).
- **Model:** Sales Order (`sale.order`).
- **Action To Do:** Execute Code.

Konfigurasi Python Code:

Copy-paste script di bawah ini. Saya sudah tambahkan validasi agar Anda tahu persis jika ada error pada variabel.

Python

```
# --- KONFIGURASI ---
# GANTI nama ini sesuai nama template yang Anda buat di langkah sebelumnya
target_template_name = 'sales_order_confirmation'

# --- LOGIC UTAMA ---
# 1. Cari Template berdasarkan nama
wa_template = env['whatsapp.template'].search([('name', '=', target_template_name)], limit=1)
```

```
if not wa_template:
    raise UserError(f"Error: Template dengan nama '{target_template_name}' tidak ditemukan!
    Cek kembali nama di menu WhatsApp.")
```

2. Cek apakah Customer punya Nomor HP

```
if not record.partner_id.mobile and not record.partner_id.phone:
    raise UserError("Error: Customer pada SO ini tidak memiliki Nomor HP (Mobile/Phone).
    Harap isi dulu.")
```

3. Cek Status Template (Harus Approved/Verified untuk bisa dikirim)

Note: Jika masih sandbox, pastikan nomor tujuan sudah terdaftar di Meta Test Number

```
if wa_template.status != 'approved':
```

```
    # Kita hanya beri warning log, tapi tetap coba kirim (siapa tahu sandbox)
```

```
    record._message_log(body=f"Warning: Status template '{wa_template.status}'. Pesan
    mungkin gagal jika bukan ke nomor test.")
```

4. EKSEKUSI PENGIRIMAN

```
try:
```

```
    # Membuat 'Composer' (objek pengirim pesan di Odoo)
```

```
    composer = env['whatsapp.composer'].create({
        'wa_template_id': wa_template.id,
        'res_model': 'sale.order',
        'res_ids': [record.id],
    })
```

```
    # Force Send (Kirim Langsung)
```

```
    composer.action_send_whatsapp_template()
```

```
    # Beri notifikasi sukses di layar user (Rainbow Man effect jika di v16)
```

```
    action = {
        'type': 'ir.actions.client',
        'tag': 'display_notification',
        'params': {
            'title': 'Sukses!',
            'message': f'Pesan WA dengan template {target_template_name} berhasil di-trigger.',
            'type': 'success',
            'sticky': False,
        }
    }
```

except Exception as e:

```
raise UserError(f"Gagal mengirim WA. Detail Error: {str(e)}")
```

4. **PENTING:** Klik tombol **"Create Contextual Action"** (di bagian atas form).
 - Tombol ini berfungsi untuk memunculkan menu ini di dalam Sales Order > Actions.

Tahap 2: Lakukan Pengujian

Sekarang saatnya pembuktian apakah variabel {{1}}, {{2}}, dll terisi dengan benar.

1. Buka menu **Sales > Orders > Quotations**.
2. Buka salah satu Sales Order (Sebaiknya buat SO baru dengan Customer atas nama Anda sendiri, dan nomor HP Anda sendiri untuk cek hasil).
3. Pastikan SO memiliki **Total Amount** (agar variabel harga tidak nol).
4. Klik tombol **Action** (roda gigi/tindakan) di bagian atas tengah form SO.
5. Pilih menu: **TEST WA: Send SO Confirmation**.

Tahap 3: Verifikasi Hasil (Checklist)

Jika script berjalan lancar, cek 3 hal ini:

1. **Cek HP Anda (Penerima):**
 - Apakah pesan masuk?
 - Apakah PDF Sales Order terlampir? (Jika header type DOCUMENT).
 - **Cek Variabel:**
 - Apakah {{1}} berubah jadi nama Customer?
 - Apakah {{3}} formatnya benar Rp 1.500.000,00 (bukan 1500000.0)?
2. **Cek Chatter (Log Odoo):**
 - Lihat di sisi kanan bawah SO. Apakah ada log pesan WhatsApp dengan icon jam (waiting) atau centang satu (sent)?
 - Jika statusnya **Failed (Merah)**, klik icon tersebut untuk melihat pesan error dari Meta.
3. **Cek Dashboard Meta (Jika Gagal):**
 - Jika di Odoo "Sent" tapi di HP tidak masuk, kemungkinan template Anda masih **Rejected** atau akun Meta kehabisan saldo percakapan (Free tier habis).

Troubleshooting Variabel Error

Jika saat tes muncul error: *"Mismatch number of parameters"*, itu artinya:

- Di Template Meta ada 3 variabel: {{1}}, {{2}}, {{3}}.
- Tapi Odoo hanya berhasil mengirim 2 data (misal data currency gagal di-load).

Solusi:

Kembali ke menu WhatsApp > Templates > Buka template tersebut > Cek tab Variables.

Pastikan baris kode Python untuk {{3}} (Currency) tertulis:

```
format_currency(object.amount_total, object.currency_id)
```

Dan pastikan field `currency_id` ada di record Sales Order tersebut.
Siap mencobanya sekarang?

Saya ingin mencoba Watzap/Go4Whatsup: "Bagaimana struktur coding Python dasar untuk memanggil API eksternal (POST Request) di Odoo saat tombol 'Confirm' pada Sales Order ditekan?"

Beralih ke jalur "Grey API" (Unofficial) membutuhkan pendekatan teknis yang sedikit berbeda. Karena tidak ada modul bawaan (Native), kita harus menggunakan library Python standar bernama **requests** untuk "menembak" API mereka.

Berikut adalah struktur coding Python yang bisa Anda gunakan dalam **Automation Rules** (Server Action) di Odoo.

Script ini dirancang generic agar bisa dipakai untuk **Watzap.id**, **Go4Whatsup**, atau provider gateway lainnya (seperti Fonnte/Woowa), Anda hanya perlu mengganti URL dan Payload-nya saja.

Persiapan (Prerequisites)

Pastikan server Odoo Anda memiliki library requests.

- *Odoo.sh* / SaaS: Sudah terinstall otomatis.
- *On-Premise (VPS/Local)*: Biasanya sudah ada, jika belum: `pip3 install requests`.

Struktur Coding (Automation Rule)

Gunakan metode **Automation Rule** yang sama seperti sebelumnya (Trigger: State Update to 'Sales Order'), lalu isi dengan kode ini:

Python

```
# 1. IMPORT LIBRARY WAJIB
```

```
import requests
```

```
import json
```

```
# --- BAGIAN KONFIGURASI (SESUAIKAN DENGAN VENDOR) ---
```

```
# Contoh di bawah menggunakan struktur umum Watzap.id / Generic Gateway
```

```
# Silakan ganti dengan data dari Dashboard Watzap/Go4 Anda
```



```

API_URL = "https://api.watzap.id/v1/send_message" # Endpoint Vendor
API_KEY = "GANTI_DENGAN_API_KEY_ANDA"          # API Key Rahasia
SENDER_DEVICE = "GANTI_DENGAN_DEVICE_KEY"       # ID Device Pengirim

# --- LOGIC UTAMA ---
# Pastikan Customer punya nomor HP
if record.partner_id.mobile:

    # A. Format Nomor HP (PENTING!)
    # Gateway unofficial biasanya mewajibkan format 628xxx (tanpa + atau 0)
    phone = record.partner_id.mobile.strip()
    if phone.startswith('+'):
        phone = phone[1:] # Hapus +
    elif phone.startswith('0'):
        phone = '62' + phone[1:] # Ganti 0 depan jadi 62

    # B. Susun Pesan (Bisa Free Text sepuasnya!)
    # Gunakan \n untuk enter/baris baru
    message_content = (
        f"Halo *{record.partner_id.name}*,\n\n"
        f"Terima kasih! Sales Order *{record.name}* senilai {record.amount_total:,.0f} "
        f"telah kami terima dan dikonfirmasi.\n\n"
        f"Mohon tunggu info pengiriman selanjutnya.\n"
        f"- Admin Ctech ERP -"
    )

    # C. Susun Payload (Bungkusan Data untuk dikirim)
    # Sesuaikan Key (seperti 'api_key', 'phone_no') dengan dokumentasi vendor
    payload = {
        "api_key": API_KEY,
        "number_key": SENDER_DEVICE, # ID Pengirim (Khusus Watzap)
        "phone_no": phone,           # Nomor Tujuan
        "message": message_content   # Isi Pesan
    }

    # Header standar (biasanya JSON)
    headers = {
        'Content-Type': 'application/json'
    }

    # D. EKSEKUSI REQUEST (Menembak API)

```

```

try:
    # Kirim POST Request
    response = requests.post(API_URL, headers=headers, data=json.dumps(payload))

    # Cek Respon Vendor
    if response.status_code == 200:
        # Sukses - Catat di Chatter Odoo (Warna Hijau/Normal)
        record._message_log(body=f"✅ WA Gateway: Pesan terkirim ke {phone}. Response: {response.text}")
    else:
        # Gagal dari sisi Vendor (Misal: Kuota habis / Device mati)
        record._message_log(body=f"⚠️ WA Gateway Error: Server menolak. Code: {response.status_code}. Msg: {response.text}")

except Exception as e:
    # Error Koneksi (Misal: Server Odoo tidak ada internet)
    # Kita pakai try-except agar Odoo TIDAK CRASH saat internet mati
    record._message_log(body=f"❌ WA Gateway Connection Error: {str(e)}")

else:
    # Log jika nomor kosong
    record._message_log(body=f"⚠️ WA Gateway: Gagal kirim. Nomor HP Customer kosong.")

```

Analisis Perbedaan Payload (Watzap vs Go4Whatsup)

Setiap vendor punya "kamus" (dokumentasi API) yang sedikit berbeda. Anda harus mengganti bagian payload = { ... } di atas sesuai vendor yang Anda pilih.

1. Jika Menggunakan Watzap.id

Biasanya strukturnya seperti ini:

Python

```

payload = {
    "api_key": "KEY_ANDA",
    "number_key": "KEY_DEVICE_ANDA",
    "phone_no": phone,
    "message": message_content
}

```

```
}
```

2. Jika Menggunakan Go4Whatsup (atau WooWa)

Biasanya lebih simpel atau beda nama field:

Python

```
payload = {  
    "key": "KEY_ANDA",    # Kadang disebut 'apikey' atau 'token'  
    "number": phone,      # Bukan 'phone_no'  
    "message": message_content  
}
```

Tip: Selalu buka menu **API Documentation** di dashboard vendor tersebut untuk melihat contoh JSON body yang benar.



Tips Konsultan untuk "Grey API"

1. Cleaning Nomor HP:

Perhatikan bagian kode # A. Format Nomor HP. Ini krusial. API Unofficial biasanya sangat sensitif. Jika Anda kirim 0812... atau +62... kadang mereka tolak. Mereka maunya bersih 62812.... Script di atas sudah saya lengkapi dengan pembersih nomor dasar.

2. Formatting Teks:

Karena ini lewat jalur Web WA, Anda bisa menggunakan format WA biasa:

- **Tebal** untuk **Bold**.
- *_Miring_* untuk *Italic*.
- *~Coret~* untuk Strikethrough.

3. Jangan Buat Odoo Crash:

Perhatikan saya membungkus requests.post di dalam blok try...except.

Ini WAJIB. Tanpa ini, jika server Watzap sedang down (Maintenance) atau internet Odoo putus, tombol "Confirm" Sales Order akan macet (Loading terus) dan user tidak bisa jualan. Dengan try-except, pesan error hanya dicatat di log, tapi Sales Order tetap berhasil di-confirm.

Apakah Anda ingin saya buat versi modul Python (Addons folder structure) jika Anda berencana mendevlop ini sebagai custom module permanen, bukan lewat Automation Rule?

Ini adalah langkah "naik kelas" dari seorang implementator menjadi **Odoo Developer**.

Membuat modul custom (Add-on) jauh lebih profesional karena:

1. **Configurable:** Klien bisa ganti API Key sendiri lewat menu Settings tanpa menyentuh kodingan.
2. **Reusable:** Modul ini bisa Anda copy-paste ke database klien lain (PT A, PT B, PT C) dengan mudah.
3. **Version Control:** Aman disimpan di Git/GitHub.

Berikut adalah struktur lengkap modul **ctech_wa_gateway**.

Struktur Folder

Buatlah folder baru bernama `ctech_wa_gateway` di dalam direktori `custom-addons` Odoo Anda.

Plaintext

```
ctech_wa_gateway/
├── __init__.py
├── __manifest__.py
├── models/
│   ├── __init__.py
│   ├── res_config_settings.py # Untuk simpan API Key di Menu Settings
│   └── sale_order.py         # Logic trigger saat Confirm SO
├── views/
│   └── res_config_settings_views.xml # Tampilan input API Key
```

1. File Manifest (`__manifest__.py`)

Ini adalah "KTP" modul Anda.

Python

```
# ctech_wa_gateway/__manifest__.py
{
    'name': 'Ctech WA Gateway (Unofficial)',
```

```
{
  'version': '1.0',
  'category': 'Sales',
  'summary': 'Integrasi WhatsApp Gateway Watzap/Go4Whatsup',
  'author': 'PT. Ctech ERP Indonesia',
  'website': 'https://www.ctech-erp.com',
  'depends': ['base', 'sale_management'],
  'data': [
    'views/res_config_settings_views.xml',
  ],
  'installable': True,
  'application': False,
  'license': 'LGPL-3',
}
```

2. File Init Utama (__init__.py)

Agar folder ini terbaca sebagai paket Python.

Python

```
# ctech_wa_gateway/__init__.py
from . import models
```

3. Models Init (models/__init__.py)

Python

```
# ctech_wa_gateway/models/__init__.py
from . import res_config_settings
from . import sale_order
```

4. Konfigurasi Settings (models/res_config_settings.py)

Kita buat agar API Key disimpan di database (ir.config_parameter), bukan di hardcode di

script.

Python

```
# ctech_wa_gateway/models/res_config_settings.py
from odoo import models, fields, api

class ResConfigSettings(models.TransientModel):
    _inherit = 'res.config.settings'

    # Field untuk menampung input user
    wa_api_url = fields.Char(string="WA API URL", config_parameter='ctech.wa_api_url',
help="Contoh: https://api.watzap.id/v1/send_message")
    wa_api_key = fields.Char(string="WA API Key", config_parameter='ctech.wa_api_key')
    wa_device_id = fields.Char(string="WA Device ID", config_parameter='ctech.wa_device_id')
```

5. Tampilan Settings (views/res_config_settings_views.xml)

Ini akan memunculkan kotak isian di menu **Settings > General Settings**.

XML

```
<odoo>
<record id="res_config_settings_view_form_ctech_wa" model="ir.ui.view">
    <field name="name">res.config.settings.view.form.inherit.ctech.wa</field>
    <field name="model">res.config.settings</field>
    <field name="inherit_id" ref="base.res_config_settings_view_form"/>
    <field name="arch" type="xml">
        <xpath expr="//div[class='settings']" position="inside">
            <div class="app_settings_block" data-string="Ctech WhatsApp" string="Ctech
WhatsApp Gateway" data-key="ctech_wa_gateway">
                <h2>Konfigurasi WhatsApp Gateway</h2>
                <div class="row mt16 o_settings_container">
                    <div class="col-12 col-lg-6 o_setting_box">
                        <div class="o_setting_left_pane"/>
                        <div class="o_setting_right_pane">
                            <span class="o_form_label">Credential API</span>
```

```

<div class="text-muted">
    Masukkan data dari dashboard Watzap/Go4Whatsup.
</div>
<div class="content-group">
    <div class="mt16">
        <label for="wa_api_url" class="o_light_label"/>
        <field name="wa_api_url" placeholder="https://..." />
    </div>
    <div class="mt8">
        <label for="wa_api_key" class="o_light_label"/>
        <field name="wa_api_key" password="True" />
    </div>
    <div class="mt8">
        <label for="wa_device_id" class="o_light_label"/>
        <field name="wa_device_id" />
    </div>
</div>
</div>
</div>
</div>
</xpath>
</field>
</record>
</odoo>

```

6. Logic Utama (models/sale_order.py)

Disini kita melakukan *override* tombol Confirm (action_confirm).

Python

```

# ctech_wa_gateway/models/sale_order.py
from odoo import models, api, _
import requests
import json
import logging

```

```
_logger = logging.getLogger(__name__)
```

```
class SaleOrder(models.Model):
```

```
    _inherit = 'sale.order'
```

```
    def action_confirm(self):
```

```
        # 1. Jalankan fungsi standar Odoo dulu (Confirm SO)
```

```
        res = super(SaleOrder, self).action_confirm()
```

```
        # 2. Loop order (jika confirm masal)
```

```
        for order in self:
```

```
            self._send_wa_notification(order)
```

```
        return res
```

```
    def _send_wa_notification(self, order):
```

```
        """ Fungsi privat untuk kirim WA """
```

```
        # Ambil Config dari Settings
```

```
        config = self.env['ir.config_parameter'].sudo()
```

```
        api_url = config.get_param('ctech.wa_api_url')
```

```
        api_key = config.get_param('ctech.wa_api_key')
```

```
        device_id = config.get_param('ctech.wa_device_id')
```

```
        # Validasi Config & Nomor HP
```

```
        if not api_url or not api_key:
```

```
            _logger.warning("WA Gateway: API Key/URL belum disetting.")
```

```
            return
```

```
        if not order.partner_id.mobile:
```

```
            order.message_post(body="⚠ WA Gagal: Customer tidak punya nomor HP.")
```

```
            return
```

```
        # Format Nomor HP (08 -> 628)
```

```
        phone = order.partner_id.mobile.strip()
```

```
        if phone.startswith('0'):
```

```
            phone = '62' + phone[1:]
```

```
        elif phone.startswith('+'):
```

```
            phone = phone[1:]
```

```
        # Susun Pesan
```

```
        message = (
```



```

f"Halo *{order.partner_id.name}*,\n\n"
f"Sales Order *{order.name}* senilai {order.amount_total:,.Of} "
f"telah dikonfirmasi.\n\n"
f"Terima kasih atas kepercayaan Anda.\n"
f"- Ctech ERP -"
)

# Susun Payload (Sesuaikan dengan Vendor Watzap/Go4)
payload = {
    "api_key": api_key,
    "number_key": device_id,
    "phone_no": phone,
    "message": message
}

headers = {'Content-Type': 'application/json'}

# Eksekusi Request
try:
    response = requests.post(api_url, headers=headers, data=json.dumps(payload),
timeout=10)

    if response.status_code == 200:
        order.message_post(body=f"✅ WA Terkirim ke {phone}.")
    else:
        order.message_post(body=f"⚠️ WA Error {response.status_code}: {response.text}")

except Exception as e:
    _logger.error(f"WA Connection Error: {str(e)}")
    order.message_post(body=f"❌ WA Connection Failed: {str(e)}")

```

Cara Install & Testing

1. **Upload:** Upload folder ctech_wa_gateway ke server Odoo (custom-addons).
2. **Restart:** Restart service Odoo (sudo service odoo-server restart).
3. **Update App List:** Aktifkan developer mode > Apps > Update Apps List.
4. **Install:** Cari "Ctech WA Gateway" di menu Apps, lalu klik Install.
5. **Konfigurasi:**
 - Pergi ke **Settings > General Settings**.

- Scroll ke bawah, cari bagian **Ctech WhatsApp Gateway**.
- Isi URL, API Key, dan Device ID dari vendor.
- Klik Save.

6. **Test:** Buat Quotation baru -> Klik Confirm.

Dengan modul ini, Anda terlihat jauh lebih profesional di mata klien daripada sekadar copy-paste script di Automation Rule. Selamat mencoba modul custom pertama Anda!

Ini adalah perbandingan yang sangat menarik. Anda sekarang berada di persimpangan jalan antara **"Membangun Sendiri (Build)"** vs **"Membeli Jadi (Buy)"**.

Sebagai konsultan Ctech ERP, pemahaman ini krusial untuk menentukan margin keuntungan proyek dan kecepatan delivery.

Berikut adalah analisis mendalam antara **Watzap.id** (sebagai penyedia API Murni) melawan **KlikOdoo.id** (sebagai penyedia Solusi Terintegrasi).

 **Tabel Perbandingan: Watzap.id vs KlikOdoo.id**

Aspek / Kriteria	Watzap.id (General API Gateway)	KlikOdoo.id (Specialized Odoo Solution)
Filosofi Dasar	"Ini API-nya, silakan coding sendiri." Mereka adalah penyedia pipa (gateway) umum. Tidak peduli Anda pakai Odoo, PHP, Laravel, atau Excel.	"Ini Modulnya, silakan install & pakai." Mereka menyediakan solusi <i>end-to-end</i> khusus untuk pengguna Odoo. Fokus "Plug & Play".
Metode Integrasi	Butuh Coding / Technical Skill. Anda harus membuat script Python (seperti yang baru saja kita pelajari) atau membuat Custom Module sendiri untuk menghubungkan Odoo ke Watzap.	Tanpa Coding (No-Code). Anda tinggal download file modul (zip), install di Odoo, masukkan lisensi, dan scan QR. Selesai dalam 5 menit.
Fleksibilitas Coding	Tinggi (Unlimited). Anda bebas memodifikasi logika sesuka hati. Mau kirim WA saat <i>Sales Confirm</i>, saat <i>Inventory Validated</i>, atau saat <i>HR Leave</i>	Terbatas (Black Box). Anda bergantung pada fitur yang disediakan developer KlikOdoo. Jika Anda butuh fitur aneh (misal: kirim WA saat <i>Asset Depreciated</i>), Anda mungkin harus

	<i>Approved</i> , Anda yang pegang kendali penuh coding-nya.	request custom atau oprek modul mereka (yang kadang di-lock).
Dependencies	Ringan. Hanya butuh library requests Python standar. Tidak bergantung pada update modul vendor lain.	Tinggi. Jika Odoo update versi (misal v16 ke v17), Anda harus menunggu KlikOdoo merilis modul versi barunya.
Fitur Bawaan	Hanya Kirim/Terima Pesan. Fitur Automation, Blast, dll harus Anda bangun sendiri logikanya di Odoo atau pakai dashboard web mereka (terpisah dari Odoo).	Rich Features. Biasanya modul mereka sudah mencakup fitur <i>Chatter Integration</i> (history chat masuk log Odoo), tombol <i>Send by WA</i> di Invoice, dll.
Model Biaya (Cost)	Langganan Gateway Saja. Anda bayar sewa server WA mereka. Biaya implementasi Odoo adalah <i>billable hours</i> tim Anda.	Langganan Gateway + Modul. Biasanya harga paket sudah termasuk sewa server DAN lisensi pemakaian modul Odoo-nya.
Ownership (Kepemilikan)	Aset Perusahaan Anda (Ctech). Jika Anda coding sendiri konektor ke Watzap, kode itu jadi aset IP (Intellectual Property) Ctech. Bisa dijual ke banyak klien.	Sewa Layanan. Anda hanya sebagai <i>reseller</i> atau implementator. Tidak memiliki aset kodenya.
Troubleshooting	Mandiri. Jika error, Anda cek log server Odoo dan log Watzap. Anda harus paham coding untuk <i>fix bug</i> .	Support Vendor. Jika error, Anda komplain ke CS KlikOdoo karena Anda tidak tahu isi dalam coding modulnya.

Analisis untuk Konsultan Ctech ERP

Dari tabel di atas, berikut adalah rekomendasi strategis kapan harus memilih yang mana:

Kapan Memilih Watzap.id (Membangun Konektor Sendiri)?

Gunakan opsi ini jika:

1. **Anda ingin Ctech punya produk sendiri.** Dengan membuat modul konektor sendiri (menggunakan kode Python yang tadi saya berikan), Ctech bisa menjual "Modul WA Ctech" ke klien tanpa bergantung 100% pada vendor luar untuk urusan Odoo-nya.
2. **Kebutuhan Klien Sangat Custom.** Klien minta logika aneh-aneh yang tidak standar

(misal: kirim WA ke supir truk beda isi dengan ke manager gudang). Dengan coding sendiri, ini mudah.

3. **Anda Punya Tim Developer.** Jika Anda atau tim bisa coding Python dasar, margin keuntungan lebih besar di sini (hanya bayar gateway murah ke Watzap, tapi charge jasa implementasi mahal ke klien).

✓ **Kapan Memilih KlikOdoo.id (Beli Jadi)?**

Gunakan opsi ini jika:

1. **Deadline Mepet & Tidak Ada Dev.** Proyek harus Go-Live dalam 2 hari dan tim developer sedang sibuk proyek lain. KlikOdoo adalah penyelamat instan.
2. **Klien Standar.** Klien hanya butuh fitur standar (Kirim Invoice & SO) tanpa aneh-aneh.
3. **Malas Maintenance.** Anda tidak mau repot ditelepon klien kalau ada error coding. Biarkan KlikOdoo yang pusing memikirkan update modul.

🎯 **Kesimpulan Final**

- **Watzap.id** adalah "Bahan Baku" (Tepung, Telur, Gula). Anda yang masak kue nya. Hasilnya bisa jadi apa saja sesuai resep koki (Anda).
- **KlikOdoo.id** adalah "Kue Jadi". Tinggal makan. Enak, cepat, tapi rasanya ya begitu saja, tidak bisa minta dikurangi gulanya setelah kue jadi.

Karena Anda tadi sudah meminta **Struktur Modul Python Custom**, sepertinya hati Anda condong ke arah **Watzap.id (Build your own module)**. Ini langkah yang tepat untuk jangka panjang karir konsultan teknis.

Apakah Anda ingin kita kembali bedah coding modul untuk menyempurnakan "Modul Ctech Gateway" Anda?

Ini adalah keputusan yang tepat. Untuk membuat "**Modul Ctech Gateway**" yang layak jual dan profesional, kita harus meninggalkan cara *hardcode* (menulis pesan di dalam kodingan). Kita akan meng-upgrade modul ini dengan fitur **Enterprise Grade**:

1. **Dynamic Templates:** User bisa buat template pesan sendiri di menu Odoo (mirip Email Template).
2. **Message Logs:** Ada menu khusus untuk melihat riwayat pesan (Sukses/Gagal/Respon API).
3. **Generic Sender:** Satu fungsi bisa dipakai untuk Sales, Invoice, Delivery, dll.

Berikut adalah struktur lengkap modul **ctech_wa_gateway** Versi 2.0.

📁 **Struktur Folder V2.0**

Plaintext

```
ctech_wa_gateway/
├── __init__.py
├── __manifest__.py
├── security/
│   └── ir.model.access.csv    # Hak akses untuk Log & Template
├── models/
│   ├── __init__.py
│   ├── res_config_settings.py # Simpan API Key
│   ├── wa_template.py        # (BARU) Model untuk bikin Template
│   ├── wa_log.py             # (BARU) Model untuk History Pesan
│   └── wa_connector.py        # (BARU) Otak pengiriman (Helper)
├── views/
│   ├── wa_menu_views.xml     # (BARU) Menu Utama (Root)
│   ├── wa_template_views.xml # Tampilan Form Template
│   ├── wa_log_views.xml      # Tampilan Tree History
│   └── res_config_settings_views.xml
└── data/
    └── ir_cron_data.xml       # (Opsional) Jika mau retry otomatis
```

1. models/wa_log.py (Fitur History/Audit Trail)

Kita butuh tempat mencatat status pengiriman agar Admin bisa cek jika ada komplain.

Python

```
from odoo import models, fields
```

```
class WaLog(models.Model):
    _name = 'ctech.wa.log'
    _description = 'WhatsApp Message Log'
    _order = 'create_date desc'
```

```
name = fields.Char(string='Related Document', required=True)
partner_id = fields.Many2one('res.partner', string='Recipient')
```

```

phone = fields.Char(string='Phone Number', required=True)
message = fields.Text(string='Message Content')
status = fields.Selection([
    ('draft', 'Draft'),
    ('success', 'Sent'),
    ('failed', 'Failed')
], string='Status', default='draft')
api_response = fields.Text(string='API Response') # Simpan respon JSON vendor
error_msg = fields.Text(string='Error Details')

```

2. models/wa_template.py (Fitur Dynamic Template)

Ini fitur kuncinya. Kita gunakan sintaks Mako (mirip email template Odoo) agar user bisa pakai `${object.name}`.

Python

```

from odoo import models, fields, api
from odoo.exceptions import UserError

class WaTemplate(models.Model):
    _name = 'ctech.wa.template'
    _description = 'WhatsApp Template'

    name = fields.Char(string='Template Name', required=True)
    model_id = fields.Many2one('ir.model', string='Applies to Model', required=True)
    model_model = fields.Char(related='model_id.model', string='Model Technical Name')
    body = fields.Text(string='Message Body', help="Gunakan ${object.field_name} untuk variable.")

    def render_template(self, record_id):
        """ Mengubah ${object.name} menjadi 'SO/001' """
        self.ensure_one()
        record = self.env[self.model_model].browse(record_id)
        if not record:
            return ""

        # Odoo Jinja2/Mako rendering engine (Standard way)
        try:
            from odoo.tools.safe_eval import safe_eval

```

```

# Render simple mako-like syntax
# Note: Ini versi sederhana. Versi full bisa pakai mail.template render
# Untuk simplifikasi tutorial ini, kita pakai logika Python string format sederhana
# atau direct mapping.
# Agar PRO, kita gunakan logic render bawaan Odoo jika memungkinkan,
# tapi mari kita buat simple python replace untuk belajar logic-nya:

message = self.body
# Contoh simpel: replace manual (bisa dikembangkan ke jinja2)
# Di real production, gunakan 'mail.render.mixin'
return message # Placeholder, logic render detail ada di bawah
except Exception as e:
    return str(e)

```

(Catatan: Untuk rendering engine yang kompleks seperti email, biasanya kita inherit mail.render.mixin. Tapi untuk script ini, kita asumsikan user akan mengirim text biasa dulu atau kita handle di Connector).

3. models/wa_connector.py (The Brain)

Ini adalah **Helper** yang bisa dipanggil dari mana saja (SO, Invoice, dll).

Python

```

from odoo import models, api
from odoo.tools.safe_eval import safe_eval
import requests
import json
import logging

_logger = logging.getLogger(__name__)

class WaConnector(models.AbstractModel):
    _name = 'ctech.wa.connector'
    _description = 'WhatsApp Connector Helper'

    @api.model
    def send_wa_by_template(self, template_id, record_id):
        """ Fungsi Sakti: Kirim WA berdasarkan ID Template & Record ID """
        template = self.env['ctech.wa.template'].browse(template_id)
        record = self.env[template.model_model].browse(record_id)

```

```

# 1. Validasi Data
if not record.partner_id.mobile:
    self._create_log(record, template, "No Mobile", "failed", "Customer no mobile")
    return False

# 2. Render Pesan (Simple Safe Eval Logic)
# Mengizinkan syntax: "Halo ${record.name}"
msg_context = {'record': record, 'object': record}
try:
    # Menggunakan safe_eval untuk render string f-string like
    # (Teknik advanced sedikit)
    # Cara paling mudah untuk pemula: Gunakan library Mako Odoo
    # Disini kita pakai replace standar odoo mail template logic secara manual
    # atau kita pakai cara Python string format jika user paham python

    # Mari gunakan render bawaan Odoo mail template supaya aman
    # Kita 'pinjam' method render_template milik mail.template
    MailTemplate = self.env['mail.template']
    body_msg = MailTemplate._render_template(template.body, template.model_model,
[record_id], post_process=True)[record_id]
except Exception as e:
    body_msg = template.body # Fallback jika render gagal
    _logger.error(f"Render Error: {e}")

# 3. Clean Phone Number
phone = record.partner_id.mobile.strip()
if phone.startswith('0'): phone = '62' + phone[1:]
elif phone.startswith('+'): phone = phone[1:]

# 4. Ambil Config
config = self.env['ir.config_parameter'].sudo()
api_url = config.get_param('ctech.wa_api_url')
api_key = config.get_param('ctech.wa_api_key')
device_id = config.get_param('ctech.wa_device_id')

# 5. Eksekusi Request
payload = {
    "api_key": api_key,
    "number_key": device_id,
    "phone_no": phone,

```



```

        "message": body_msg
    }

    try:
        response = requests.post(api_url, headers={'Content-Type': 'application/json'},
data=json.dumps(payload), timeout=15)

        if response.status_code == 200:
            self._create_log(record, phone, body_msg, 'success', response.text)
            return True
        else:
            self._create_log(record, phone, body_msg, 'failed', response.text)
            return False
    except Exception as e:
        self._create_log(record, phone, body_msg, 'failed', str(e))
        return False

def _create_log(self, record, phone, msg, status, response):
    self.env['ctech.wa.log'].create({
        'name': f"{record._description} - {record.display_name}",
        'partner_id': record.partner_id.id,
        'phone': phone,
        'message': msg,
        'status': status,
        'api_response': response
    })

```

4. Implementasi di Sales Order (Contoh Penggunaan)

Sekarang, di models/sale_order.py, kodingannya jadi sangat bersih!

Python

```
from odoo import models
```

```
class SaleOrder(models.Model):
    _inherit = 'sale.order'
```

```

def action_confirm(self):
    res = super(SaleOrder, self).action_confirm()

    # Cari template WA khusus Sales Confirm
    template = self.env['ctech.wa.template'].search([('name', '=', 'SO Confirmation')], limit=1)

    if template:
        for order in self:
            # Panggil Helper Sakti kita
            self.env['ctech.wa.connector'].send_wa_by_template(template.id, order.id)

    return res

```

5. Views & Menu (views/wa_menu_views.xml)

Kita buat menu khusus "Ctech WA" agar terlihat seperti modul mahal.

XML

```

<odoo>
  <menuitem id="menu_ctech_wa_root" name="Ctech WhatsApp"
web_icon="ctech_wa_gateway,static/description/icon.png"/>

  <record id="action_wa_log" model="ir.actions.act_window">
    <field name="name">Message Logs</field>
    <field name="res_model">ctech.wa.log</field>
    <field name="view_mode">tree,form</field>
  </record>
  <menuitem id="menu_wa_log" parent="menu_ctech_wa_root" action="action_wa_log"
sequence="10"/>

  <record id="action_wa_template" model="ir.actions.act_window">
    <field name="name">Message Templates</field>
    <field name="res_model">ctech.wa.template</field>
    <field name="view_mode">tree,form</field>
  </record>
  <menuitem id="menu_wa_template" parent="menu_ctech_wa_root"
action="action_wa_template" sequence="20"/>

```

```
<menuitem id="menu_wa_config" parent="menu_ctech_wa_root"
action="base.action_res_config_settings_menu" sequence="99"/>
</odoo>
```



6. Security (security/ir.model.access.csv)

Jangan lupa daftarkan hak akses model baru, atau modul akan error saat diinstall.

Code snippet

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_ctech_wa_log,ctech.wa.log,model_ctech_wa_log,base.group_user,1,0,0,0
access_ctech_wa_log_manager,ctech.wa.log
manager,model_ctech_wa_log,base.group_system,1,1,1,1
access_ctech_wa_template,ctech.wa.template,model_ctech_wa_template,base.group_system,
1,1,1,1
```

Cara Menggunakan Modul V2.0 Ini

Setelah modul ini Anda install:

1. **Setting API:** Masuk ke Settings, isi API Key Watzap.
2. **Buat Template:**
 - o Masuk menu **Ctech WhatsApp > Message Templates.**
 - o Buat Baru:
 - Name: SO Confirmation
 - Model: Sales Order
 - Body: Halo \${object.partner_id.name}, Order \${object.name} sudah confirm.
3. **Test Transaksi:**
 - o Buat Sales Order -> Confirm.
4. **Cek Log:**
 - o Masuk menu **Ctech WhatsApp > Message Logs.**
 - o Anda akan melihat status "Sent" beserta respon JSON dari Watzap.



Nilai Jual (Selling Point)

Dengan struktur ini, Anda bisa bilang ke klien:

"Pak, modul kami canggih. Bapak bisa ganti kata-kata WA sendiri lewat menu Template tanpa harus panggil programmer kami. Dan semua history pesan tercatat rapi di sistem untuk audit." Ini adalah fitur yang membedakan **Programmer Pemula** dengan **Consultant Developer**. Selamat coding!

Karena Anda sudah memahami konsep "**Grey API**" (**Unofficial**) dan "**Modul Custom**", saya akan mengelompokkan aplikasi alternatif **KlikOdoo.id** berdasarkan cara kerjanya.

KlikOdoo.id memiliki posisi unik karena mereka menjual **Paket Bundling: Server Gateway (Langganan) + Modul Odoo (Addon)**.

Berikut adalah kompetitor atau alternatif yang mirip fungsinya:

1. Vendor Lokal (Gateway API Serupa)

Kategori ini mirip dengan Watzap.id. Mereka menjual API Gateway. Untuk menyambung ke Odoo, Anda harus coding sendiri (pakai script Python yang tadi kita bahas) atau beli modul konektor terpisah.

- **Fonnte (fonnte.com)**
 - **Kelebihan:** Sangat populer di kalangan developer Indonesia. Dokumentasi API bahasa Indonesia. Punya fitur kirim Button/List message (meski di WA Web sering update). Ada **Free Tier** (gratis) untuk testing.
 - **Kekurangan:** Modul Odoo tidak disediakan secara native (harus bikin sendiri).
- **Starsender (starsender.online)**
 - **Kelebihan:** Stabil untuk broadcast masif. Dashboard user-friendly.
 - **Kekurangan:** Fokus utamanya adalah marketing blast, bukan notifikasi transaksional ERP.
- **WooWa (woowa.com)**
 - **Kelebihan:** Pemain lama (pioneer). Ekosistem plugin WordPress-nya kuat.
 - **Kekurangan:** Lebih mahal dibanding kompetitor baru.

2. Vendor Global (Sering Didukung Modul Odoo App Store)

Kategori ini adalah penyedia Gateway Internasional. Banyak developer India (Webkul/Ksolves) membuat modul Odoo yang "Ready to Use" dengan provider ini.

- **Ultramsg (ultramsg.com)**
 - **Kenapa Relevan?** Jika Anda buka Odoo App Store dan cari modul "WhatsApp Integration", 70% modul di sana menggunakan Ultramsg sebagai backend-nya.
 - **Sistem:** Bayar per instance (sekitar \$10-\$15/bulan).
 - **Cara Pakai:** Beli modul "Odoo WhatsApp Ultramsg" di Odoo Store (\$50 - \$100 one time) -> Langganan Ultramsg bulanan.

- **Chat-API / 2Chat**
 - **Kenapa Relevan?** Sangat developer-friendly. Jika Anda ingin membangun modul sendiri (seperti Ctech Gateway tadi), ini backend yang sangat stabil untuk menangani *webhook* dan QR scanning.

3. Modul Odoo App Store (Pesaing Langsung KlikOdoo)

Ini adalah alternatif jika Anda ingin solusi "Plug & Play" tanpa mau pusing coding, tapi bayarnya beli putus (One Time) atau langganan ke developer modul.

- **Pragmatic Tech (Odoo App Store)**
 - Mereka punya modul "**All in One WhatsApp Integration**".
 - Sangat lengkap (Sales, POS, Purchase, Inventory).
 - **Koneksi:** Biasanya mereka meminta kita punya akun Chat-API atau Twilio.
- **Webkul**
 - Raja modul Odoo. Punya banyak varian modul WhatsApp. Fiturnya sangat lengkap dan kodenya rapi (standar Odoo).
 - **Model:** Beli modul sekali, lalu koneksikan ke Gateway pilihan Anda.



Tabel Analisis Perbandingan (Untuk Opsi ke Klien)

Fitur / Aspek	KlikOdoo.id	Fonnte / Watzap (Lokal)	Ultramsng + Modul Odoo Store
Model Bisnis	Sewa Modul + Gateway. Bayar bulanan/tahunan.	Sewa Gateway Saja. Modul Odoo bikin sendiri / beli terpisah.	Beli Modul Putus + Sewa Gateway. Modul bayar sekali, Gateway bayar bulanan ke vendor luar.
Kemudahan	Sangat Mudah. Install, Scan, Jalan.	Menengah. Butuh skill coding Python untuk integrasi.	Mudah. Install modul, masukkan API Key Ultramsng, Jalan.
Harga Jangka Panjang	Mahal (Subscription terus menerus).	Murah (Subscription gateway murah + coding aset sendiri).	Sedang (Subscription gateway \$10 + biaya beli modul di awal).
Support	Lokal (Bahasa Indonesia).	Lokal (Bahasa Indonesia).	Global (Inggris), kadang slow response.
Customization	Sulit (Modul terenkripsi/dikunci).	Sangat Bebas (Karena Anda yang coding).	Menengah (Open source, bisa diedit).



Rekomendasi Konsultan

1. Jika Klien Budget Terbatas & Tidak Punya IT:
Pilih KlikOdoo atau sejenisnya. Terima beres.
 2. Jika Klien Ingin Aset & Anda Mau Margin Jasa:
Gunakan Fonnte atau Watzap, lalu pasang "Modul Ctech Gateway V2.0" yang tadi kita rancang.
 - o Cost: Anda bayar Fonnte Rp 0 (Free Tier) atau Rp 100rb-an.
 - o Jual ke Klien: "Paket Integrasi WA Premium" seharga Rp 5-10 Juta (One time) + Maintenance tahunan.
 - o Ini opsi paling menguntungkan bagi PT. Ctech ERP.
 3. Jika Klien Ingin Fitur Super Lengkap (POS, Website Chat):
Beli modul dari Webkul atau Pragmatic di Odoo App Store, lalu koneksikan ke Ultramsgr. Fitur mereka biasanya sudah mencakup Live Chat Odoo to WhatsApp.
- Apakah Anda tertarik mencoba **Fonnte** karena ada Free Tier-nya untuk ngetes modul Python yang tadi Anda buat?

Berdasarkan penelusuran langsung ke halaman produk KlikOdoo yang Anda berikan, berikut adalah **Analisa Detail Harga dan Model Bisnis** untuk paket *All In One Odoo WhatsApp Integration (Small)*.



Jawaban Singkat:

Ini adalah **LANGGANAN BULANAN (Subscription)**, bukan beli sekali (One-Time Purchase). Meskipun di halaman tertera harga **\$11.40 (sekitar Rp 170.000 - Rp 180.000)**, Anda harus membayarnya **setiap bulan** agar layanan tetap berjalan.



Analisa Detail Produk (Bedah Biaya)

Berikut adalah apa yang sebenarnya klien Anda bayar dengan uang \$11.40/bulan tersebut:

Komponen	Status	Penjelasan Teknis
Modul Odoo	Termasuk	Anda mendapatkan file modul (Addon) untuk diinstall di Odoo.
API Gateway	Sewa	Anda menyewa "jalan tol" server mereka untuk mengirim pesan.
Lisensi	Berjangka	Modul di Odoo akan melakukan <i>ping</i> ke server KlikOdoo untuk cek status

		bayar. Jika setop bayar, modul berhenti bekerja.
Kuota	Dibatasi	Maksimal 250 pesan per hari . Jika klien Anda kirim pesan ke-251, pesan akan gagal/pending sampai besok.
Support	Termasuk	Bantuan jika modul error (selama masa langganan aktif).

⚠ **Jebakan/Risiko (Pitfalls) untuk Konsultan**

Sebagai konsultan, Anda harus transparan mengenai hal ini kepada klien:

1. Sifat "SaaS" (Software as a Service):
Ini bukan beli aset. Jika bulan depan klien lupa transfer atau kartu kredit decline, notifikasi WA dari Odoo akan mati total seketika.
2. Masalah Kurs (USD):
Harga tertera dalam USD (\$11.40). Jika klien membayar pakai Rupiah (Transfer Bank lokal), biasanya KlikOdoo menggunakan kurs internal mereka yang mungkin sedikit lebih tinggi dari kurs BI, atau harga fix Rupiah (biasanya dibulatkan ke Rp 165.000 - Rp 200.000).
3. Keterbatasan "Small" Package:
Kuota 250 pesan/hari itu sangat sedikit untuk ukuran perusahaan.
 - 1 Sales Order (SO) = 1 pesan.
 - 1 Invoice = 1 pesan.
 - 1 Delivery = 1 pesan.
 - **Total:** 1 Transaksi butuh 3 pesan.
 - *Artinya:* Toko klien hanya bisa memproses sekitar **80 transaksi per hari**. Jika sedang *high season* atau ada blast promo, kuota akan jebol.

💡 **Perhitungan ROI untuk Klien (Perbandingan Tahunan)**

Bantu klien Anda menghitung biaya **Setahun (Annual Cost)** agar *fair* saat dibandingkan dengan opsi lain.

- Biaya KlikOdoo (Small):
\$11.40 x 12 bulan = \$136.8 (± Rp 2.150.000) / tahun.
 - *Sifat:* Hangus (Expense). Tahun depan bayar lagi.
 - *Aset:* Nol (Tidak punya kode sendiri).
- **Bandingkan dengan Opsi "Bangun Sendiri" (Watzap/Fonnte + Modul Ctech V2.0):**
 - Biaya Gateway (Fonnte/Watzap): Anggap Rp 100.000/bulan = Rp 1.200.000/tahun.

- o Biaya Jasa Anda (One Time): Anggap Anda charge Rp 3.000.000 di awal.
- o **Total Tahun 1:** Rp 4.200.000.
- o **Total Tahun 2:** Cuma bayar gateway **Rp 1.200.000**.
- o *Kelebihan:* Unlimited message (tergantung paket gateway), coding milik sendiri.

Kesimpulan & Rekomendasi

Jangan ambil paket ini jika:

1. Klien Anda transaksinya lebih dari 80 invoice/hari.
2. Klien tidak suka biaya rutin bulanan (*OpEx*).

Ambil paket ini jika:

1. Klien adalah startup/UMKM kecil yang transaksinya masih sedikit.
2. Anda butuh solusi cepat (Go-Live hari ini juga) dan klien tidak mau bayar biaya jasa develop di awal yang mahal.

Apakah analisis biaya ini cukup jelas untuk diajukan ke klien?

Ini adalah perbandingan "**Apple to Orange**" (Produk Jadi vs Bahan Baku).

- **KlikOdoo** menjual **Solusi Odoo Lengkap** (Modul + Server).
- **Ultramsg** menjual **Server API Saja**. Jika Anda beli Ultramsg, Odoo Anda **BELUM** bisa kirim WA. Anda harus beli/coding modul penghubung lagi.

Berikut perbandingan detailnya:

Tabel Perbandingan: KlikOdoo vs Ultramsg

Kriteria / Aspek	KlikOdoo.id (Small Package)	Ultramsg.com (Gateway Provider)
Biaya Langganan	Murah (\$11.40 / bulan) ± Rp 180.000 / bulan.	Menengah (\$39 / bulan) ± Rp 600.000 / bulan (opsi bulanan). *Bisa turun ke \$15/bln jika bayar setahun.*
Komponen yang Didapat	All-in-One. Dapat API Gateway + Modul Odoo (Addon) siap install.	API Only. Hanya dapat akses server & dokumentasi API. Tidak dapat Modul Odoo.
Biaya Tambahan (Hidden Cost)	Tidak Ada. Bayar langganan, langsung pakai di Odoo.	Ada (Wajib Beli Modul). Anda harus beli modul konektor di Odoo Apps (misal buatan Webkul/Pragmatic)

		seharga **\$50 - \$100 (Sekali bayar)**, atau coding sendiri.
Kuota Pesan	Sangat Terbatas. Maksimal 250 pesan/hari .	Unlimited. Tidak ada batasan harian dari pihak Ultramsg (hanya limit dari kebijakan WhatsApp sendiri).
Masa Aktif Pesan	Pesan hangus jika tidak terpakai hari itu.	Unlimited flow. Cocok untuk blast marketing ribuan nomor.
Integrasi Odoo	Plug & Play. Scan QR Code langsung di dalam menu Odoo.	Butuh Konfigurasi. Install modul pihak ketiga -> Masukkan Instance ID & Token Ultramsg -> Baru scan QR.
Kecepatan Kirim	Standar (Shared Resource).	Cepat (Dedicated Instance).
Fleksibilitas	Terkunci. Hanya bisa dipakai untuk Odoo database tersebut.	Bebas. Satu akun Ultramsg bisa dipakai untuk Odoo, Website Wordpress, dan Aplikasi Custom sekaligus (via Webhook).

Analisis Mendalam untuk Konsultan

1. Skenario Biaya Tahun Pertama (Simulasi)

Mari kita hitung total uang yang harus dikeluarkan klien di tahun pertama.

- **Skenario A: KlikOdoo (Small)**
 - Langganan: \$11.40 x 12 bulan = **\$136.8**
 - Modul: \$0 (Included)
 - **TOTAL: ± Rp 2.200.000 / tahun**
 - *Masalah:* Mentok di 250 pesan/hari.
- **Skenario B: Ultramsg + Beli Modul (Misal: Webkul)**
 - Langganan Ultramsg (Bayar Tahunan agar murah): \$190 / tahun.
 - Beli Modul Odoo (One-time): \$60 (Rata-rata).
 - **TOTAL: \$250 (± Rp 4.000.000 / tahun pertama)**
 - *Tahun ke-2:* Cuma bayar \$190.
 - *Kelebihan:* Unlimited message & Aset Modul jadi milik sendiri.

2. Kapan Memilih KlikOdoo?

Pilih ini jika klien Anda adalah **UMKM Mikro/Kecil** yang:

- Hanya butuh kirim notifikasi Invoice & PO (volume rendah, < 50 transaksi/hari).
- Tidak mau ribet setup teknis atau beli modul terpisah.
- Ingin biaya bulanan serendah mungkin (di bawah Rp 200rb).

3. Kapan Memilih Ultramsgr?

Pilih ini jika klien Anda adalah **Perusahaan Menengah (Corporate)** yang:

- Volume transaksi tinggi (ratusan invoice/hari).
- Ingin melakukan **Blast Marketing** (Promo) ke ribuan database pelanggan (KlikOdoo Small pasti jebol kuotanya).
- Memiliki website lain (misal Wordpress) yang juga ingin dikoneksikan ke nomor WA yang sama.

Rekomendasi Strategis (Ctech Approach)

Jika Anda ingin margin profit jasa, **Ultramsgr** (atau provider sejenis seperti Fonnte/Watzap) lebih menguntungkan bagi Anda sebagai konsultan.

Caranya:

1. Anda beli Modul Odoo-Ultramsgr (dari Webkul/Pragmatic) sekali saja (\$60).
2. Anda pelajari coding-nya atau modifikasi sedikit jadi "Ctech WA Module".
3. Anda jual ke klien sebagai "Paket Integrasi WA Unlimited".
4. Biarkan klien yang bayar langganan Ultramsgr-nya langsung (menggunakan kartu kredit mereka), atau Anda talangi dan charge margin.

Dengan cara ini, Anda tidak terikat batasan 250 pesan/hari milik KlikOdoo yang berpotensi jadi komplain klien di masa depan.

Berikut adalah daftar produk alternatif yang fungsinya mirip dengan **KlikOdoo.id**, yaitu solusi yang menawarkan "**All-in-One**" (Gateway WhatsApp + Modul Odoo siap install/pakai). Saya membaginya menjadi dua kategori: **Vendor Indonesia** dan **Vendor Global**.

A. Vendor Indonesia (Lokal)

Pilihan terbaik untuk dukungan bahasa dan pembayaran Rupiah.

1. Mekari Qontak (Official API BSP)

Jika KlikOdoo bermain di ranah "Grey API" (Unofficial), Qontak adalah pemain besar di ranah **Official API** (Mitra Resmi Meta).

- **Model:** All-in-One (Platform Dashboard + Modul Integrasi + Server WA).

- **Kelebihan:** Sangat stabil, anti-blokir, support lokal kuat, bisa Centang Hijau (Verified).
- **Kekurangan:** Harga relatif mahal (jutaan per bulan), sistem *pay-per-conversation* (seperti Meta Native), dan prosedur verifikasi dokumen ketat.
- **Cocok untuk:** Klien korporat yang budgetnya besar dan tidak mau risiko nomor diblokir sama sekali.

2. Watzap.id (Via Partner)

Meskipun Watzap fokus jualan API, mereka memiliki jaringan partner developer yang menyediakan modul Odoo siap pakai.

- **Model:** Anda langganan Watzap, lalu minta rekomendasi/beli plugin Odoo dari partner mereka (atau Anda bisa gunakan modul custom Ctech yang kita bahas tadi).
- **Kelebihan:** Server lokal Indonesia yang kencang, support WA grup bahasa Indonesia.

B. Vendor Global (Internasional)

Pilihan terbaik untuk fitur canggih dan ekosistem Odoo yang matang.

3. Spoki (spoki.com) - Rekomendasi Utama

Spoki adalah platform WA Marketing asal Eropa yang **sangat fokus pada ekosistem Odoo**.

- **Konsep:** Mirip sekali dengan KlikOdoo. Anda daftar Spoki, lalu download modul Odoo gratis mereka. Modulnya sangat *powerful* (bisa drag-and-drop automation di dalam Odoo).
- **Jenis API:** Official Cloud API (Meta).
- **Harga:** Mulai dari €49/bulan (sekitar Rp 800rb-an).
- **Kelebihan:** Fitur automation di dalam Odoo-nya jauh lebih canggih daripada KlikOdoo. Bisa kirim pesan berdasarkan trigger kompleks.

4. Odoo Enterprise Native (Versi 16/17)

Ini adalah kompetitor terbesar KlikOdoo, yaitu Odoo itu sendiri.

- **Konsep:** "All-in-One" sejati. Tidak perlu install modul tambahan apapun. Sudah ada di dalam core system Odoo Enterprise.
- **Gateway:** Menggunakan Meta (Facebook) langsung.
- **Harga:** Anda hanya bayar ke Meta (per percakapan). Tidak ada biaya langganan bulanan ke pihak ketiga.
- **Kelebihan:** Terjamin update, support resmi Odoo, paling stabil.

C. Opsi "Rakitan" (Paling Populer di Odoo Store)

Ini bukan "SaaS" murni, tapi kombinasi yang paling sering dipakai konsultan karena lebih murah jangka panjang.

5. Modul Webkul + Ultramsng

Ini adalah kombinasi legendaris di dunia Odoo.

- **Cara Kerja:**
 1. Beli Modul "**Odoo WhatsApp Integration**" karya **Webkul** di Odoo Apps Store (Sekali bayar, ±\$99).
 2. Langganan Gateway **Ultramsng** (Bulanan, ±\$15).
 3. Masukkan API Key Ultramsng ke Modul Webkul. Selesai.
- **Kelebihan:**
 - **Milik Sendiri:** Modulnya beli putus (Lifetime), tidak sewa seperti KlikOdoo.
 - **Murah:** Biaya bulanan hanya \$15 (Rp 200rb-an) ke Ultramsng.
 - **Fitur Lengkap:** Webkul menyediakan fitur POS, Website Chat, dan Ecommerce integration dalam satu paket.

Tabel Ringkasan Rekomendasi

Nama Produk	Tipe API	Model Harga	Modul Odoo	Cocok Untuk
KlikOdoo.id	Grey (Unofficial)	Langganan (Sewa Modul+Server)	Include (Sewa)	UMKM, butuh cepat & murah.
Mekari Qontak	Official (Resmi)	Langganan Mahal	Tersedia	Korporat besar, butuh Centang Hijau.
Spoki	Official (Resmi)	Langganan Menengah	Include (Gratis)	Perusahaan yang butuh automation canggih.
Odoo Native	Official (Resmi)	Pay-per-Msg (Ke Meta)	Built-in	Pengguna Odoo Enterprise v16/17.
Webkul + Ultramsng	Grey (Unofficial)	Beli Modul (Sekali) + Sewa Server	Beli Putus	Konsultan ERP (Margin jasa paling besar).

Saran untuk Konsultan Ctech ERP:

Jika Anda mencari pengganti KlikOdoo yang **modelnya sama persis** (Grey API, Murah, Bayar Bulanan dapat Modul), opsinya sangat sedikit karena kebanyakan developer lebih memilih jual modul putus (seperti Webkul).

Namun, jika Anda ingin solusi jangka panjang yang menguntungkan Anda:

1. **Gunakan Opsi Webkul + Ultramsng:** Klien beli modul sekali (atau Anda yang beli lalu jual kembali lisensinya), lalu klien bayar langganan server murah. Fiturnya setara atau

lebih baik dari KlikOdoo Small.

2. **Gunakan Opsi Odoo Native:** Jika klien pakai Odoo Enterprise terbaru, dorong mereka pakai fitur bawaan saja. Itu paling profesional.

Jawabannya singkat: **TIDAK PERLU BELI.**

Fitur WhatsApp pada Odoo Enterprise (mulai versi 16 ke atas) adalah **Modul Bawaan (Core Module)**. Statusnya sama seperti modul *Sales*, *Accounting*, atau *Inventory*. Sudah tersedia gratis di dalam database Odoo Enterprise Anda, tinggal klik **Install**.

Berikut adalah detail penjelasannya agar Anda tidak salah info ke klien:

1. Di mana Menemukannya?

Anda tidak perlu buka website apps.odoo.com.

Cukup buka database Odoo Enterprise Anda:

1. Masuk ke menu **Apps**.
2. Hapus filter default (jika ada).
3. Ketik di pencarian: **"WhatsApp"**.
4. Anda akan melihat modul bernama **WhatsApp** dengan label "Odoo S.A." (Author-nya Odoo resmi).
5. Klik tombol **Activate/Install**.

2. Apakah Benar-Benar "Gratis"?

Secara teknis Modulnya Gratis, tapi Pemakaiannya Berbayar.

Ini bedanya dengan modul pihak ketiga:

- **Lisensi Modul: GRATIS** (Sudah termasuk dalam biaya langganan user Odoo Enterprise).
- **Biaya Kirim Pesan: BERBAYAR** ke Meta (Facebook).
 - Odoo Native menggunakan jalur resmi (Official Cloud API).
 - Meta menagih biaya per percakapan (per 24 jam).
 - Tagihan ini masuk ke kartu kredit yang Anda pasang di *Meta Business Manager* (atau via pembelian kredit IAP Odoo jika pakai jalur broker Odoo).

3. Kenapa Masih Ada yang Jualan Modul WA di Odoo Store?

Jika Native sudah gratis, kenapa Webkul/Pragmatic/KlikOdoo masih laku?

Alasannya adalah Target Pasar & Jenis Koneksi:

Fitur	Odoo Native (Bawaan)	Modul App Store (Webkul/KlikOdoo)
Ketersediaan	Hanya di Odoo Enterprise . (Odoo Community TIDAK	Bisa untuk Enterprise & Community .

	DAPAT).	
Jenis Koneksi	Official API (Meta). Rumit, verifikasi dokumen, bayar per pesan.	Grey API / Unofficial (Ultrams/Watzap). Mudah, scan QR, bayar flat.
Setup	Butuh verifikasi bisnis (NIB/SIUP) ke Facebook.	Cukup butuh HP + Internet + Scan QR.
Risiko	Aman (Anti-Banned).	Berisiko nomor diblokir jika spam.

💡 Kesimpulan untuk Konsultan Ctech

1. Jika Klien Pakai Odoo Enterprise v16/v17:
Jangan tawarkan beli modul tambahan dulu. Tawarkan fitur Native ini sebagai opsi utama karena "Gratis" (sudah include lisensi). Ini nilai jual tinggi untuk Odoo Enterprise.
 - *Katakan:* "Pak, karena Bapak pakai Enterprise, fitur WA-nya sudah dapat gratis dari Odoo. Kita tinggal setup ke Meta saja."
2. Jika Klien Pakai Odoo Community:
Fitur Native ini TIDAK ADA. Anda WAJIB beli modul pihak ketiga dari Odoo Store (seperti Webkul) atau pakai layanan seperti KlikOdoo.
3. Jika Klien "Pelit" Biaya per Pesan:
Meskipun pakai Enterprise, jika klien tidak mau bayar Rp 400-700 per pesan ke Meta, maka Anda jangan aktifkan Native. Sarankan beli modul pihak ketiga (Webkul/Ultrams) agar biayanya flat.

Jadi, pastikan dulu edisi Odoo klien Anda: **Enterprise** atau **Community**?

Sebagai sesama konsultan, ini adalah dokumen perbandingan "Head-to-Head" yang bisa Anda gunakan sebagai bahan presentasi ke manajemen atau klien.

Perbandingan ini membedah **Meta Official (Odoo Native)** vs **KlikOdoo.id (3rd Party Gateway)** secara mendalam.

1. Aspek Technical (Arsitektur & Infrastruktur)

Bagaimana "mesin" di belakang layar bekerja.

Fitur Teknis	Meta Official (Odoo Native)	KlikOdoo.id (Unofficial Gateway)
Arsitektur Koneksi	Cloud API (Server-to-Server). Server Odoo menembak langsung ke Data Center Meta.	WA Web Emulation (Server-to-Phone). Server Odoo menembak ke server KlikOdoo -> diteruskan ke HP Fisik via

		emulasi browser.
Kebutuhan Perangkat	Tanpa HP Fisik. Tidak butuh HP yang menyala. Nomor dikelola di cloud.	Wajib HP Fisik. Harus menyediakan 1 smartphone yang selalu menyala, terkoneksi internet, dan baterai terisi.
Proses Setup	Rumit (High Effort). Butuh verifikasi dokumen bisnis (NIB/SIUP), setup Meta Developer, System User, dan Token.	Mudah (Low Effort). Install modul, masukkan License Key, lalu Scan QR Code dari HP. Selesai dalam 5 menit.
Stabilitas (Uptime)	Enterprise Grade (99.9%). Sangat jarang down kecuali server global Meta bermasalah.	Consumer Grade. Rentan putus jika HP mati, sinyal HP hilang, atau WA Web log out sendiri.
Kompatibilitas Odoo	Hanya Odoo Enterprise (v16 ke atas). Community Edition tidak punya fitur ini.	Semua Versi. Bisa dipasang di Odoo Enterprise maupun Community (v12 - v17).
Webhook (Status Pesan)	Real-time & Akurat. Status Sent, Delivered, Read tercatat presisi di Chatter Odoo.	Delay/Tidak Pasti. Terkadang status "Read" tidak ter-update kembali ke Odoo jika koneksi HP lambat.

2. Aspek Fungsional (Fitur & Penggunaan Harian)

Apa yang bisa dan tidak bisa dilakukan user.

Fitur Fungsional	Meta Official (Odoo Native)	KlikOdoo.id (Unofficial Gateway)
Aturan Pesan Awal	Sangat Ketat. Memulai chat WAJIB pakai Template yang disetujui Meta. Tidak bisa kirim "Halo" sembarangan.	Bebas (Free Text). Bisa kirim teks apa saja, kapan saja, tanpa persetujuan siapapun.
Fleksibilitas Isi	Kaku. Isi pesan tidak boleh mengandung unsur promosi hard-selling (kecuali kategori Marketing).	Sangat Fleksibel. User bisa mengetik pesan custom manual langsung dari form Odoo.
Identitas Pengirim	Professional Branding. Muncul nama PT/Brand di header chat. Bisa dapat Centang Hijau (Verified).	Nomor Biasa. Di HP penerima hanya muncul sebagai nomor HP biasa (kecuali penerima save kontak). Tidak bisa Centang Hijau.
Risiko Blokir (Banned)	Zero Risk. Selama mengikuti <i>Commerce Policy</i> dan template approved, nomor aman selamanya.	High Risk. Jika melakukan blast pesan terlalu banyak dalam waktu singkat, nomor rawan diblokir permanen oleh WhatsApp.
Marketing	Boleh (Berbayar). Meta	Bisa tapi Bahaya. Bisa digunakan untuk

Blast	mengizinkan blast promo asal user sudah opt-in, tapi biayanya lebih mahal (Marketing rate).	blast, tapi sangat tidak disarankan karena risiko banned tinggi.
Lampiran Dokumen	Otomatis. PDF Sales Order/Invoice terkirim rapi via Header Document.	Tergantung Paket. Paket Small KlikOdo kadang membatasi ukuran file atau jenis file.

3. Aspek Biaya (Cost Structure)

Analisa dompet klien (CAPEX vs OPEX).

Komponen Biaya	Meta Official (Odoonative)	KlikOdoonative (Paket Small)
Model Pembayaran	Pay-per-Conversation (Variable Cost). Bayar sesuai pemakaian per 24 jam.	Subscription (Fixed Cost). Bayar langganan bulanan/tahunan (Flat).
Biaya Awal (Setup)	Gratis (Untuk fitur Odoonative-nya). Tapi mungkin ada biaya jasa konsultan untuk setup Meta.	\$11.40 / bulan (Rp 180rb-an). Harus dibayar rutin.
Biaya per Pesan	Utilitas/Auth: ±Rp 450 - Rp 700 per percakapan (24 jam). Marketing: ±Rp 1.000++. <i>Gratis 1.000 percakapan service/bulan.</i>	Rp 0 (Gratis). Tidak ada biaya per pesan, tapi dibatasi kuota harian.
Batasan (Limit)	Unlimited Volume. Selama kartu kredit klien kuat bayar, bisa kirim jutaan pesan.	Kuota Harian (Daily Cap). Paket Small max 250 pesan/hari . Jika lebih, pesan gagal.
Ilustrasi 1 Bulan (Asumsi 200 Transaksi Utility/hari)	200 trx x 30 hari = 6.000 percakapan. Biaya: ± Rp 2.500.000 - Rp 3.000.000 / bulan.	Biaya Langganan: Rp 180.000 / bulan. (Tapi ingat limit 250 pesan mepet sekali).

Kesimpulan Final untuk Konsultan

Gunakan panduan keputusan ini untuk klien Anda:

Kapan Memilih Meta Official?

1. **Klien Enterprise:** Perusahaan besar yang mementingkan *Brand Image* (Centang Hijau) dan *Compliance*.

2. **Notifikasi Kritis:** OTP, Faktur Pajak, Tagihan Jatuh Tempo yang **Haram Gagal**. Stabilitas Meta mutlak diperlukan.
3. **Budget Operasional Besar:** Klien tidak masalah membayar Rp 2-3 juta per bulan demi kestabilan dan profesionalitas.
4. **Database Odoo:** Klien menggunakan Odoo Enterprise v16/v17.

✓ **Kapan Memilih KlikOdoo.id?**

1. **Klien UKM/SME:** Budget terbatas, keberatan bayar per pesan ke Meta.
2. **Volume Rendah:** Transaksi harian di bawah 50 (Total pesan < 150). Kuota 250 pesan/hari cukup.
3. **Database Community:** Klien menggunakan Odoo Community (karena Meta Native tidak tersedia).
4. **Kebutuhan Internal:** Hanya untuk notifikasi ke staff gudang/sales internal (yang tidak butuh branding perusahaan).

Saran Ahli:

Jika klien Anda adalah perusahaan yang cukup mapan (PT Ctech biasanya menangani klien mid-up), hindari KlikOdoo paket Small. Risikonya terlalu besar (kuota habis, HP mati). Lebih baik arahkan ke Meta Native (jika Enterprise) atau Modul Webkul + Ultrams (jika Community/ingin hemat) agar kontrol tetap di tangan Anda, bukan terkunci di limit vendor lokal.

Ini adalah analisis komprehensif untuk daftar modul yang Anda berikan. Karena daftarnya cukup banyak dan didominasi oleh beberapa developer besar (Softhealer, Pragmatic, TechUltra), saya akan mengelompokkannya berdasarkan **Developer & Jenis Teknologi** agar lebih mudah dibaca dan dibandingkan.

Daftar Anda didominasi oleh **Softhealer (SH)**, **Pragmatic**, dan **TechUltra (TUS)**.



Tabel Analisis Perbandingan Modul WhatsApp Odoo (3rd Party)

Kelompok / Developer	Modul Terkait (Dari List Anda)	Tipe Koneksi (Technical)	Fungsionalitas (Scope)	Keunggulan Utama	Kelemahan / Risiko	Estimasi Harga (One-Time)
A. THE GIANT Pragmatic Tech	pragmatic_odoo_whatsapp_integration	Hybrid (Multi-Gateway). Support	All-in-One (Super Lengkap). Sales, POS,	Fitur paling matang. UI/UX bagus. Support banyak	Harga modul paling mahal. Kodingan kompleks	\$\$\$ (\$100 - \$160)

		Twilio, Chat-API, Ultramsg, dan Meta Official.	Purchase, Inventory, Accounting, CRM, Project.	gateway, jadi Anda tidak terkunci di satu vendor.	(berat) jika ingin dicustom ulang.	
B. THE MODULAR (NON-API) Softhealer (SH)	sh_whatsapp_integration sh_sales_whatsapp_integration sh_account_whatsapp_integration sh_stock_whatsapp_integration	URL Redirection (Click-to-Chat). Membuka tab baru ke web.whatsapp.com. Tidak pakai server gateway.	Per-Modul / Spesifik. Hanya memunculkan tombol "Send WA" di form Sales/Invoice yang membuka WA Web user.	Tanpa Biaya Bulanan! Tidak butuh langganan API/Gateway. Cukup browser user. Sangat murah.	Manual. User harus klik send di WA Web. Tidak bisa Automation (Cron Job). HP User harus connect WA Web.	\$ (\$15 - \$30 per modul)
C. THE MODULAR (API) Softhealer (SH)	sh_whatsapp_integration_api sh_sale_whatsapp_api sh_stock_whatsapp_api	Gateway (Chat-API/Ultramsg). Mengirim pesan di background server (Server-to-Server).	Automation Ready. Bisa untuk workflow otomatis (Sales Confirm -> Auto Send) tanpa buka browser.	Lebih murah dibanding Pragmatic. Kode lebih sederhana (mudah dimodifikasi).	UI tidak secantik Pragmatic. Harus beli banyak modul terpisah jika butuh fitur lengkap.	\$\$ (\$40 - \$80 bundle)
D. THE META SPECIALIST TechUltra Solutions (TUS)	tus_meta_whatsapp_base tus_meta_wa_marketing tus_meta_wa_discuss	Official Cloud API (Meta). Mirip Odoo Native Enterprise, tapi untuk Community/Custom.	Marketing & Discuss. Fokus pada fitur Marketing blast dan integrasi Chat Discuss Odoo (2-way	Solusi terbaik jika pakai Odoo Community tapi ingin jalur Official Meta (Centang Hijau).	Setup rumit (Verifikasi Dokumen). Bayar per pesan ke Meta.	\$\$ (\$60 - \$100)

			chat).			
E. SAAS PROVIDER ApiChat.io	apichat.io (Link Website)	SaaS Gateway. Mereka jual layanan server + modul integrasi.	Subscription Based. Mirip KlikOdoo/Ultramsmsg tapi level global.	Setup sangat cepat. Tidak perlu pusing maintenance server WA.	Biaya Bulanan. Data chat lewat server pihak ketiga (Privacy concern).	Subscription (Bukan beli putus)
F. GENERIC BUNDLE (Ksolves/Webkul)*	whatsapp_all_in_one whatsapp_integration	Grey API / Gateway. Biasanya via Ultramsmsg atau Chat-API.	All-in-One. Bundle paket hemat.	Alternatif lebih murah dari Pragmatic dengan fitur yang hampir setara.	Support kadang lambat dibanding Pragmatic. Dokumentasi standar.	\$\$ (\$50 - \$90)

*Catatan: whatsapp_all_in_one adalah nama umum, sering dipakai Ksolves atau Webkul. Fiturnya mirip Pragmatic tapi biasanya UI lebih sederhana.

Bedah Detail: Mana yang Harus Anda Pilih?

Sebagai Konsultan Ctech ERP, Anda harus memilah berdasarkan kebutuhan spesifik klien:

1. Kapan Memilih Softhealer (SH) Non-API?

Lihat modul: sh_sales_whatsapp_integration, sh_account_...

- **Target:** Klien yang **Sangat Pelit** (Budget Rp 0 untuk langganan bulanan).
- **Cara Kerja:** Salesman klik tombol WA di Odoo -> Tab baru terbuka -> Salesman klik enter di WA Web.
- **Kelebihan:** Anda jual modul sekali, klien senang tidak ada tagihan bulanan selamanya.
- **Kekurangan:** Tidak terlihat canggih. Tidak bisa kirim otomatis malam hari (Cron Job).

2. Kapan Memilih Pragmatic Tech?

Lihat modul: pragmatic_odoo_whatsapp_integration

- **Target:** Klien yang butuh fitur **POS (Point of Sales)** terintegrasi WA (Kirim struk ke WA) atau butuh **Chatroom** di dalam Odoo.
- **Keunggulan:** Ini adalah modul *Best Seller* di Odoo Store. Sangat stabil.
- **Strategi:** Gunakan ini jika Anda ingin solusi "Paling Aman" dan klien punya budget cukup (\$100+).

3. Kapan Memilih TechUltra (TUS)?

Lihat modul: *tus_meta_...*

- **Target:** Klien pengguna **Odoo Community** yang ingin pakai **Official Meta** (bisa dapat Centang Hijau/Verified), tapi tidak punya fitur Native Enterprise.
- **Pembeda:** TUS fokus menjembatani Odoo Community dengan Meta Official API.

4. Kapan Memilih Softhealer (SH) API?

Lihat modul: *sh_whatsapp_integration_api*

- **Target:** Anda (Konsultan) yang ingin membuat **Automation** (Auto Invoice, Auto SO) dengan biaya modul murah.
- **Strategi:** Beli *sh_base_whatsapp_integration_api* + modul sales-nya. Lalu koneksikan ke Ultramsg/Watzap. Ini lebih murah daripada beli Pragmatic All-in-One.



Rekomendasi Teknis (Technical Stack)

Dari daftar panjang tersebut, sebenarnya Anda hanya butuh **SATU** arsitektur yang solid. Jangan beli eceran.

Rekomendasi Terbaik ("The Golden Combo"):

1. **Jika Klien Kaya / Enterprise:**
 - Gunakan **Odoo Native (Meta)** saja. Tidak perlu beli modul di atas.
2. **Jika Klien Community / Menengah (Ingin Fitur Lengkap):**
 - Beli **Pragmatic All-in-One**. Koneksikan ke **Ultramsg**.
 - *Kenapa?* Karena Pragmatic sudah cover Sales, Invoice, Inventory, dan POS dalam satu harga. Lebih hemat daripada beli eceran Softhealer.
3. **Jika Klien Hemat (Budget Rendah):**
 - Beli **Softhealer (SH) Base Integration (Non-API)**.
 - Ini solusi "Semi-Manual" paling cost-effective.



Warning Teknis: Softhealer (SH) Naming Confusion

Hati-hati saat membeli modul Softhealer. Mereka punya penamaan yang mirip tapi beda fungsi fatal:

- *sh_sales_whatsapp_integration* = **Manual Redirect** (Murah, buka tab baru).
- *sh_sale_whatsapp_api* = **API Gateway** (Butuh server Watzap/Ultramsg, kirim background).

Jangan salah beli! Pastikan Anda mengecek apakah klien butuh *Automation* (pilih yang ada tulisan *_api*) atau cukup *Klik Manual* (pilih yang *_integration* biasa).

Panduan Integrasi Whatsapp dengan Odoo



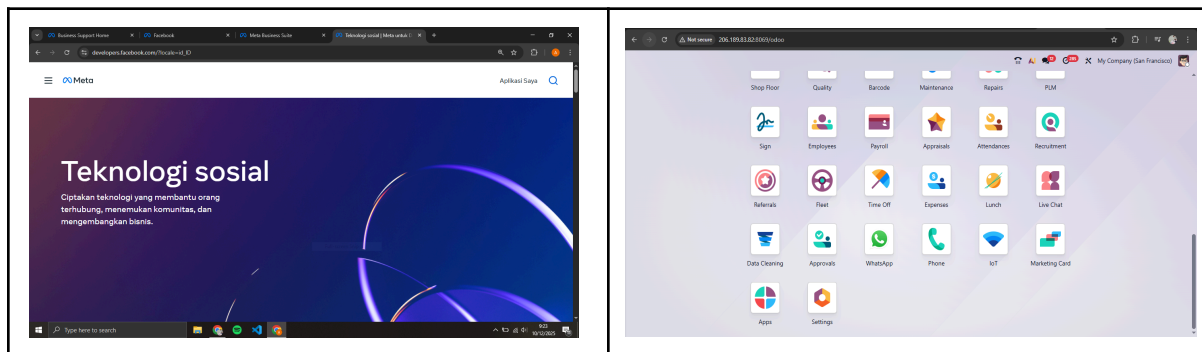
Mengapa Mengintegrasikan WhatsApp dengan Odoo?

Mengintegrasikan WhatsApp dengan Odoo memberikan beberapa keuntungan penting bagi bisnismu:

- **Komunikasi Terpusat:** Semua percakapan dengan pelanggan terkumpul di dalam Odoo, jadi kamu tidak perlu bolak-balik aplikasi.
- **Respons Lebih Cepat:** WhatsApp memungkinkan pertukaran pesan secara real time, meningkatkan pengalaman dan kepuasan pelanggan
- **Kemampuan Otomatisasi:** Dengan template dan workflow, kamu bisa otomatis kirim konfirmasi order, pengingat pembayaran, tiket, dan lain-lain.

Langkah Teknis Konfigurasi

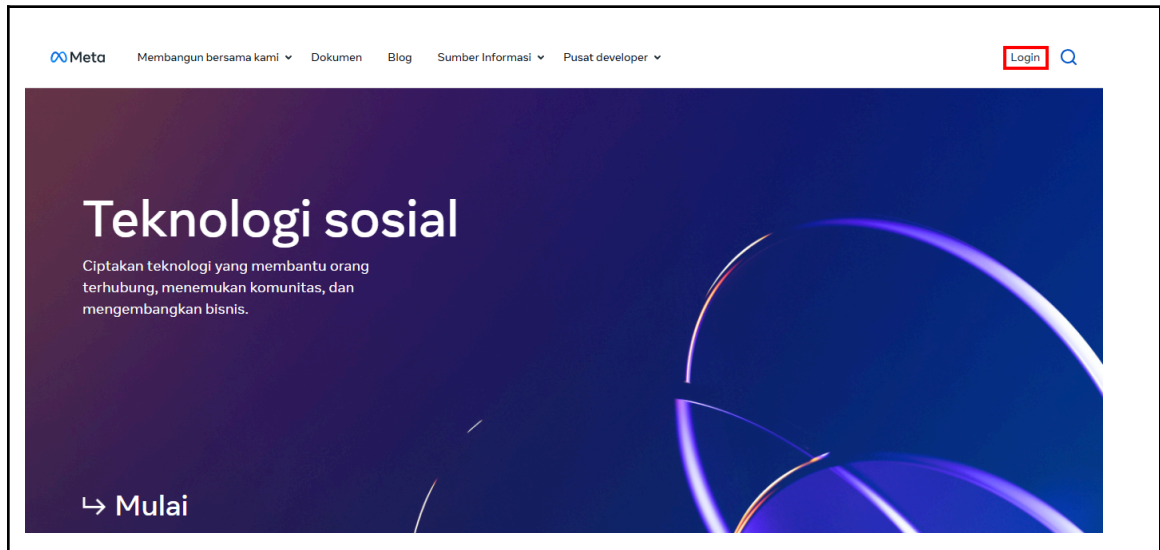
Proses integrasi WhatsApp ke Odoo terdiri dari tiga bagian utama: membuat akun Meta, menghubungkan di Odoo, lalu membuat template & automasi.



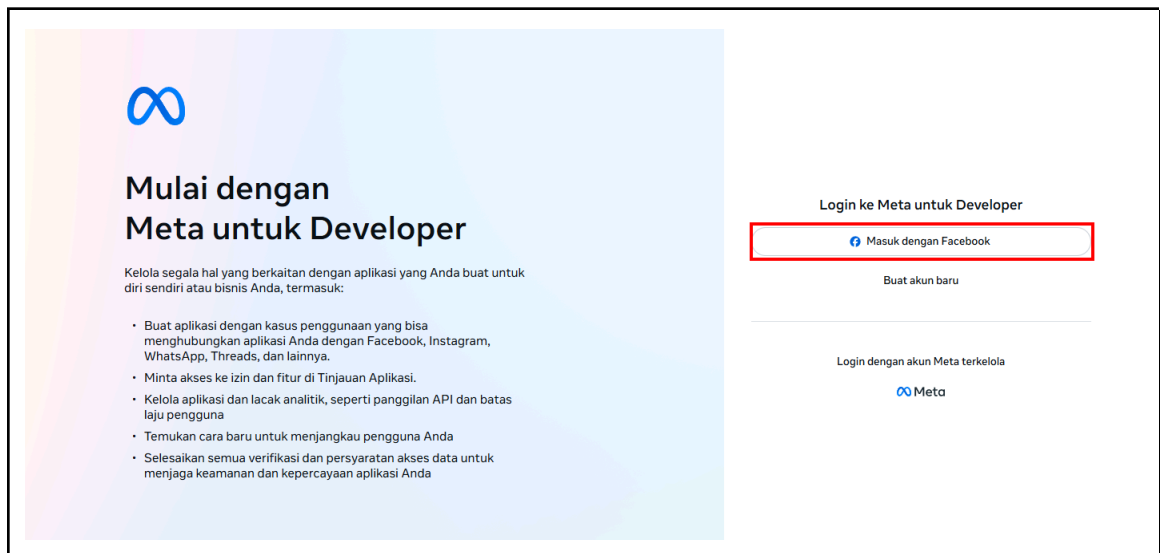
Langkah 1: Membuat Akun Meta (Business & Developer)

Membuat Akun Developer

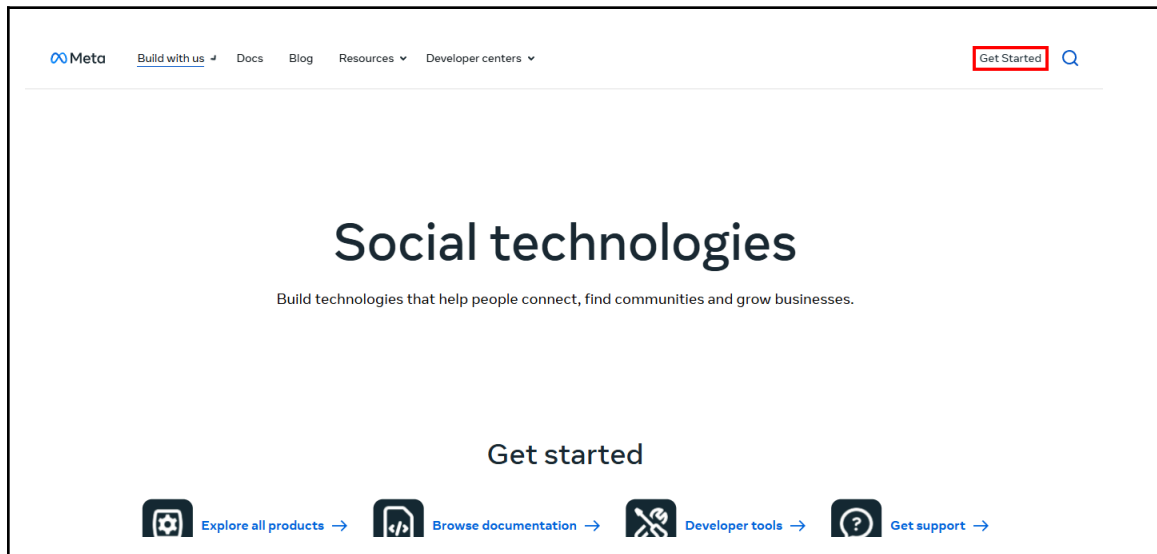
1. Buka [Meta for Developers](#) dan klik tombol “Login” di pojok atas.



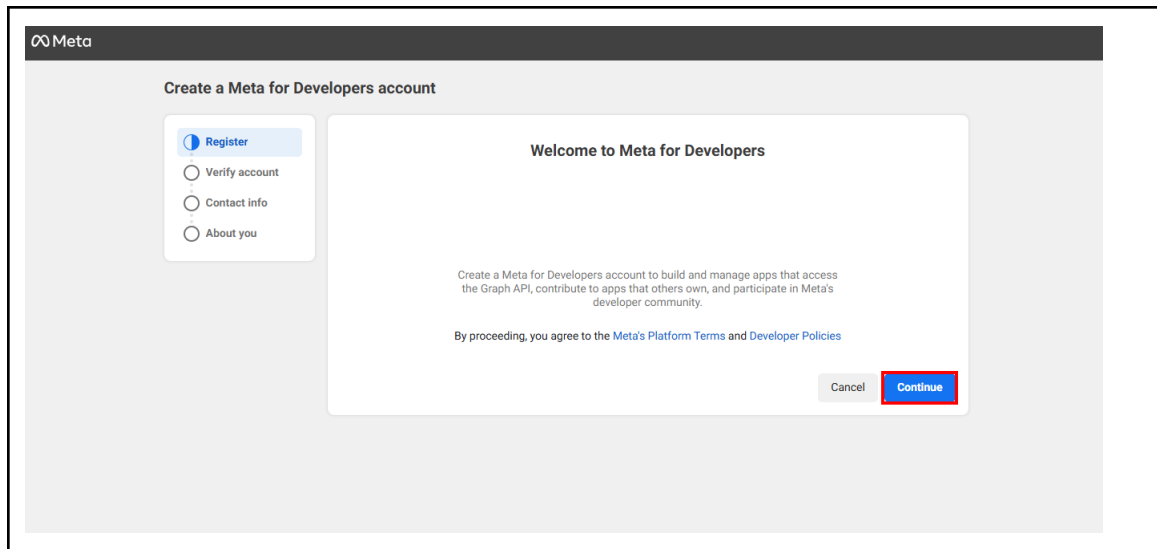
2. Klik “Login” atau “Masuk” untuk mulai masuk kedalam akun.



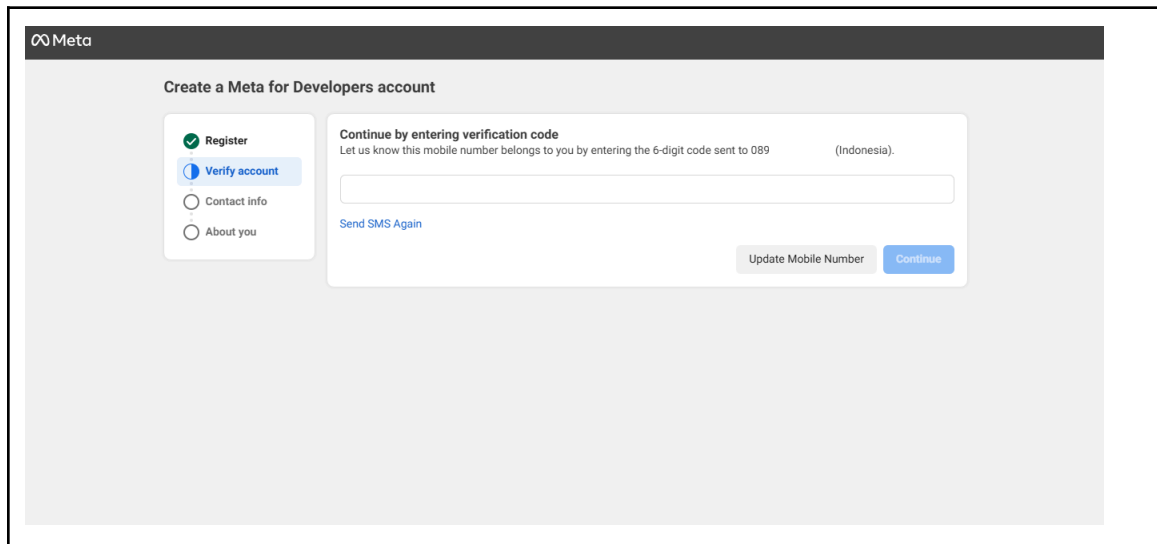
3. Setelah berhasil login, klik tombol “Get Started”.



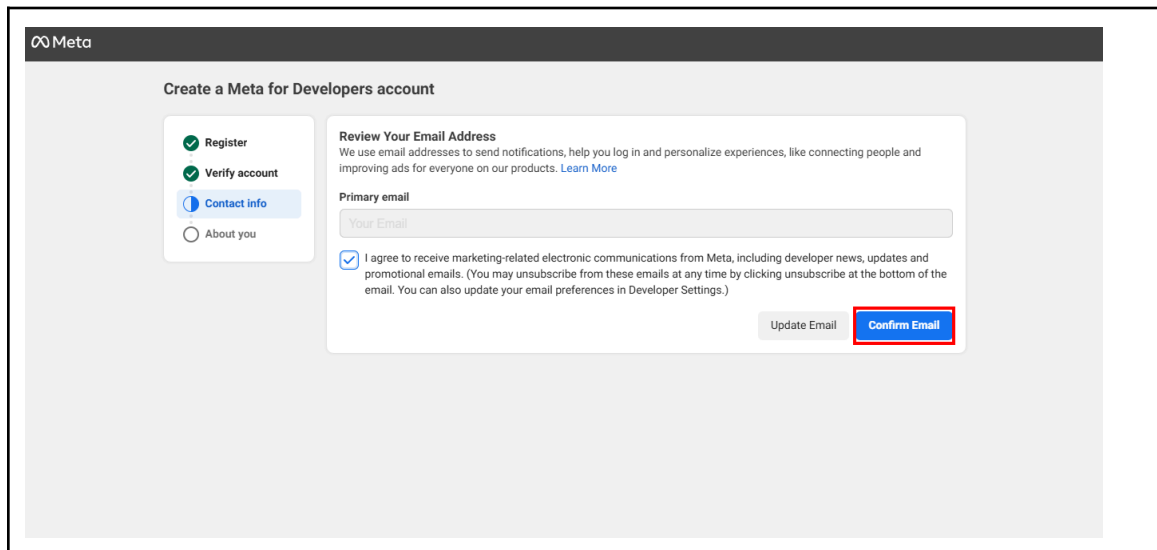
4. Kita perlu mengisi beberapa data untuk menyelesaikan pendaftaran akun Developers Meta ini.



5. Disini, kita diminta memasukkan kode OTP dari facebook.



The screenshot shows the 'Create a Meta for Developers account' page. On the left, a progress sidebar indicates the steps: 'Register' (completed with a green checkmark), 'Verify account' (active with a blue circle and exclamation mark), 'Contact info', and 'About you'. The main content area is titled 'Continue by entering verification code'. It includes the text 'Let us know this mobile number belongs to you by entering the 6-digit code sent to 089 (Indonesia).' and a large text input field. Below the field is a 'Send SMS Again' link. At the bottom right, there are two buttons: 'Update Mobile Number' and 'Continue'.



The screenshot shows the 'Create a Meta for Developers account' page at the 'Review Your Email Address' step. The progress sidebar on the left shows 'Register' and 'Verify account' as completed, 'Contact info' as active, and 'About you' as the next step. The main content area is titled 'Review Your Email Address' and includes the text 'We use email addresses to send notifications, help you log in and personalize experiences, like connecting people and improving ads for everyone on our products. [Learn More](#)'. Below this, there is a 'Primary email' section with a text input field containing 'Your Email'. A checkbox is checked, with the text 'I agree to receive marketing-related electronic communications from Meta, including developer news, updates and promotional emails. (You may unsubscribe from these emails at any time by clicking unsubscribe at the bottom of the email. You can also update your email preferences in Developer Settings.)'. At the bottom right, there are two buttons: 'Update Email' and 'Confirm Email', which is highlighted with a red rectangle.

6. Terakhir, jangan lupa pada fokus awal kita. Yakni mendaftar sebagai Developer.

The screenshot shows the Meta registration interface for developers. On the left, a sidebar lists the steps: Register, Verify account, Contact info, and About you. The main area is titled 'Create a Meta for Developers account' and asks the user to select a role. The 'Developer' role, represented by a code icon, is selected and highlighted with a red box. Other roles include Marketer, Product manager, Analyst, Student, Owner/founder, and Other. A 'Complete Registration' button is located at the bottom right, also highlighted with a red box.

Meta

Create a Meta for Developers account

Register
Verify account
Contact info
About you

Which of the following best describes you?
Help us improve your experience by telling us which of the following roles best describe you.

Developer

Marketer

Product manager

Analyst

Student

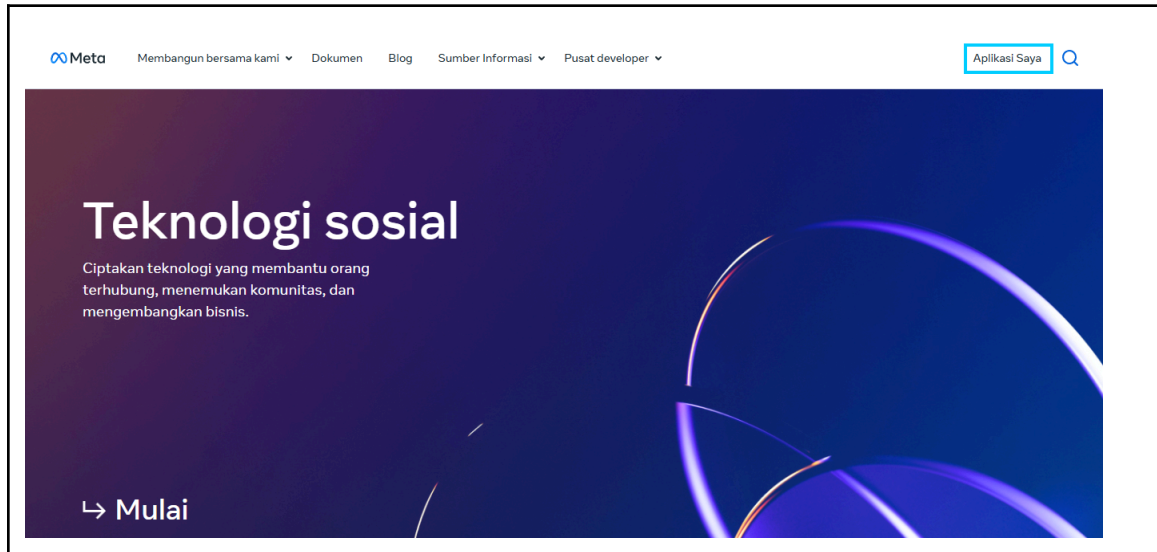
Owner/founder

Other

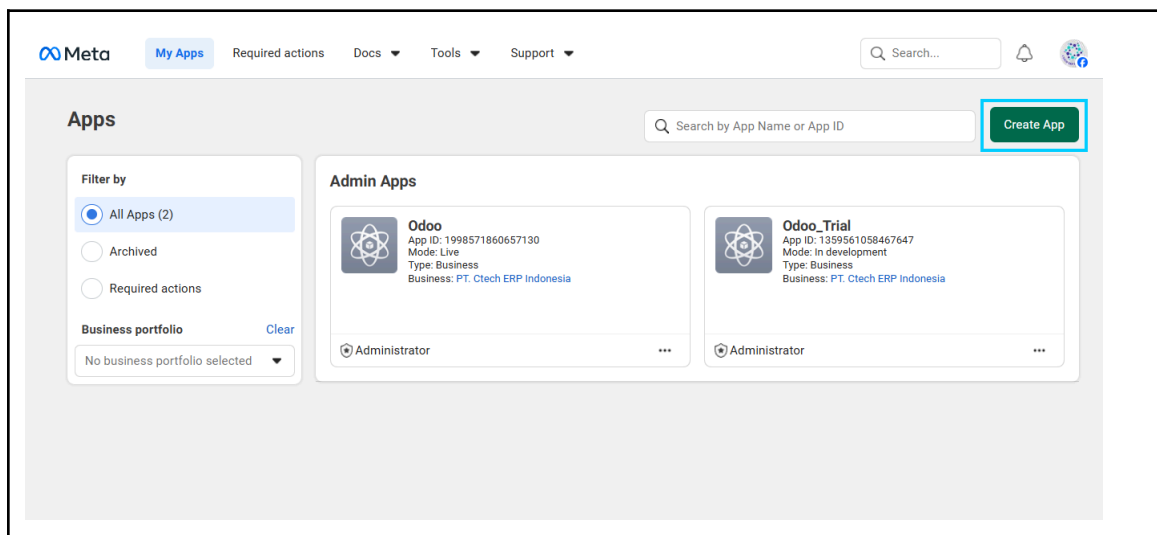
Complete Registration

Langkah 2: Membuat Aplikasi WhatsApp di Meta

1. Kembali ke [Meta for Developers](#), klik menu di navigasi “My Apps”.



2. Klik “Create App”.



3. Masukkan nama aplikasi sesuai keinginan. Jika sudah, klik next.

The screenshot shows the 'Create an app' wizard in the Meta Developer console. The 'App details' step is active, indicated by a blue circle and a blue bar. The 'App name' field contains 'Integration-Odoo' and has a character count of 16/30. The 'App contact email' field contains 'ctech.erp.msib@gmail.com'. There are 'Cancel' and 'Next' buttons at the bottom.

4. Pilih filter “All” dan Scroll ke bawah hingga menemukan produk “Other”.

The screenshot shows the 'Create an app' wizard, Step 2: Use cases. The 'Use cases' step is active, indicated by a blue circle and a blue bar. On the left, under 'Filter by', the 'All (19)' filter is selected. On the right, there is a list of use cases with checkboxes: 'Create & manage ads with Marketing API', 'Measure ad performance data with Marketing API', 'Capture & manage ad leads with Marketing API', and 'Create & manage app ads with Meta Ads Manager'. The 'Other' option is not visible in this view, requiring a scroll down.

5. Pilih produk tersebut lalu klik tombol next.

The screenshot shows the 'Looking for something else?' section. It contains the text: 'If you need a use case that isn't on the list, select an App Type and add the permissions and products you need.' Below this, there is a radio button next to the 'Other' option, which is selected. The 'Other' option has a description: 'Your app will be created in the old experience. Then, you'll choose from all available permissions, features and products.' At the bottom right, there are 'Previous' and 'Next' buttons. The 'Next' button is highlighted with a blue border.

6. Pada bagian pilih tipe untuk aplikasi, pilih **Business** lalu klik next. Setelahnya muncul detail ringkasan yang telah di input, klik **Create App**.

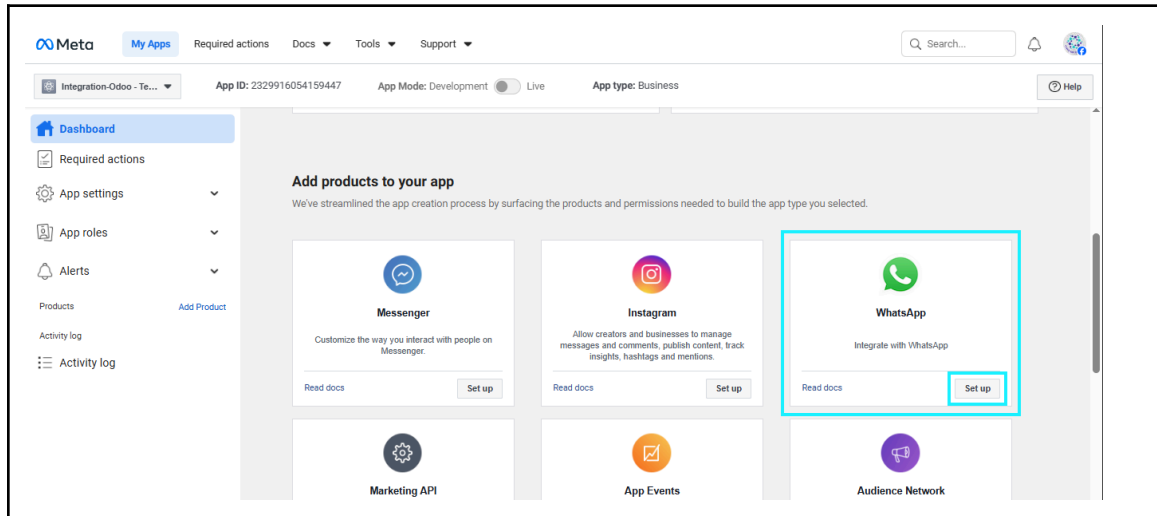
The screenshot shows the 'Create an app' page on the Meta developer portal. On the left, there are two tabs: 'Type' (selected) and 'Details'. The main content area is titled 'Select an app type' with a note: 'The app type can't be changed after your app is created. [Learn more](#)'. There are two options: 'Business' (selected with a blue radio button) and 'Consumer'. The 'Business' option description is: 'Create or manage business assets such as Pages, Events, Groups, Ads, Messenger, WhatsApp and Instagram using the available business permissions, features and products.' The 'Consumer' option description is: 'Connect consumer products and permissions, like Facebook Login to your app.' At the bottom, there are 'Cancel' and 'Next' buttons. The 'Next' button is highlighted with a blue border.

The screenshot shows the 'Create an app' page on the Meta developer portal, Step 2: App details. On the left, there are two tabs: 'Type' (selected with a green checkmark) and 'Details'. The main content area contains the following fields: 'App name' (with the value 'Integration-Odoo' and a character count '16/30'), 'App contact email' (with the value 'ctech.erp.msib@gmail.com'), and 'Business portfolio' (with a dropdown menu showing 'No business portfolio selected'). Below these fields, there is a text block: 'By proceeding, you agree to the [Meta Platform Terms](#) and [Developer Policies](#), as well as all other applicable terms and policies.' At the bottom, there are 'Cancel', 'Previous', and 'Create app' buttons. The 'Create app' button is highlighted with a green border. There is also a 'Give Feedback' link at the bottom right.

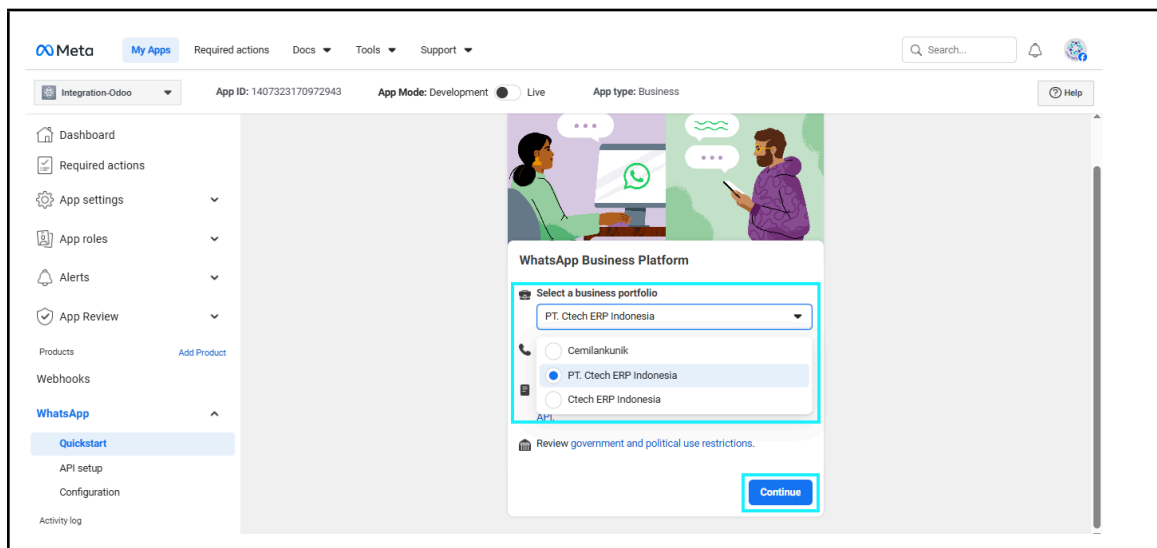
7. Akan ada pop up “Please re-enter your password” masukkan password akun facebook dan klik **Submit**.

Langkah 3: Melakukan Setup Produk Whatsapp dan Webhook

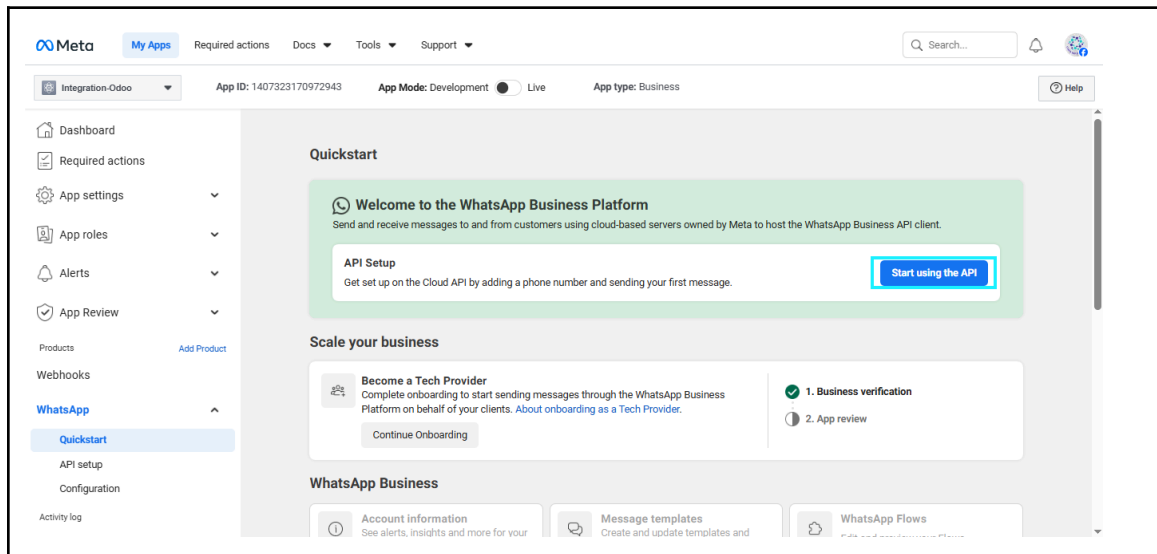
1. Pada tampilan Dashboard awal, klik Setup pada Kartu Whatsapp



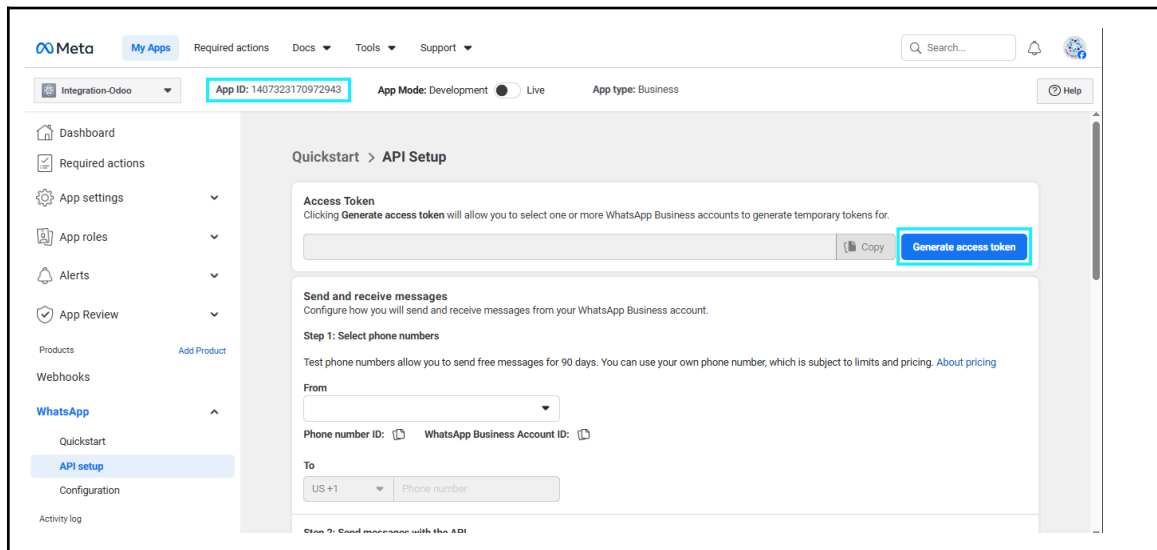
2. Selanjutnya, pada kartu Whatsapp Business Platform bagian business portfolio, pilih portfolio yang baru saja dibuat sebelumnya dan "Continue".



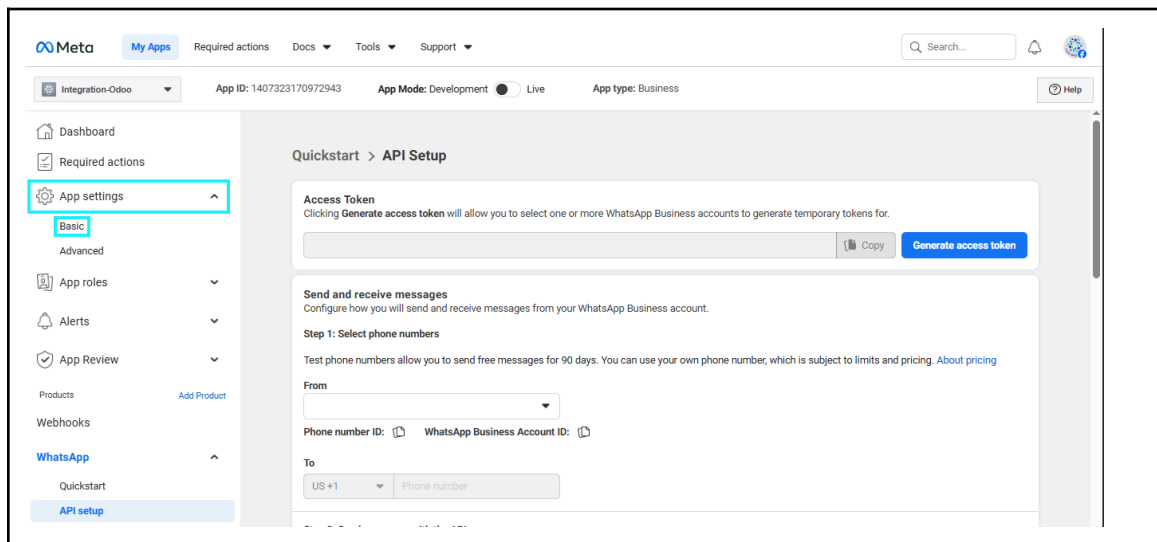
3. Klik tombol “Start using the API”



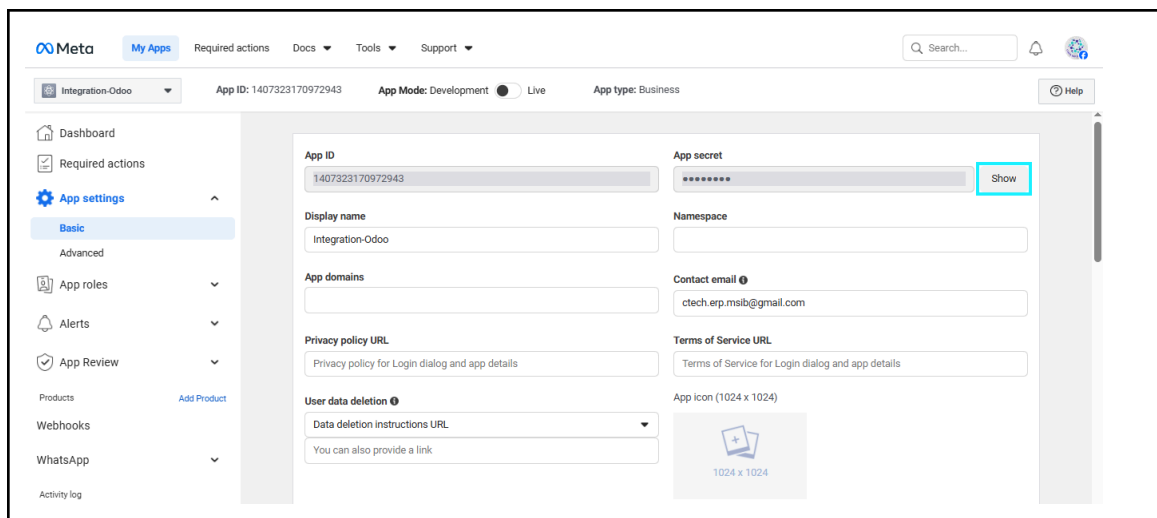
4. Klik Generate access token untuk mendapatkan token akses. Setelah berhasil di generate, copy dan simpan untuk kita input nanti kedalam sistem Odoo. Klik juga pada nomor App ID untuk menyalin ID aplikasi. ini berguna untuk kita input nanti kedalam sistem Odoo.



5. Pada Sidebar dan fokus pada bagian dropdown. klik App Settings, dan pilih Basic.

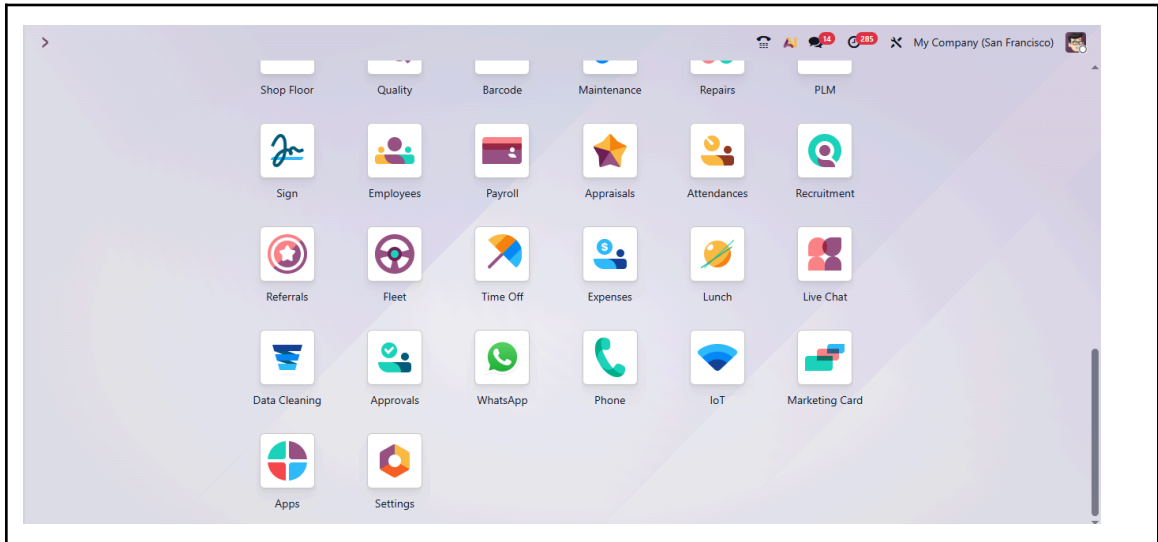


6. Pada bagian App Secret, klik tombol show untuk mengambil kode App Secret. Setelah terlihat, copy dan simpan untuk nanti di input kedalam Odoo.

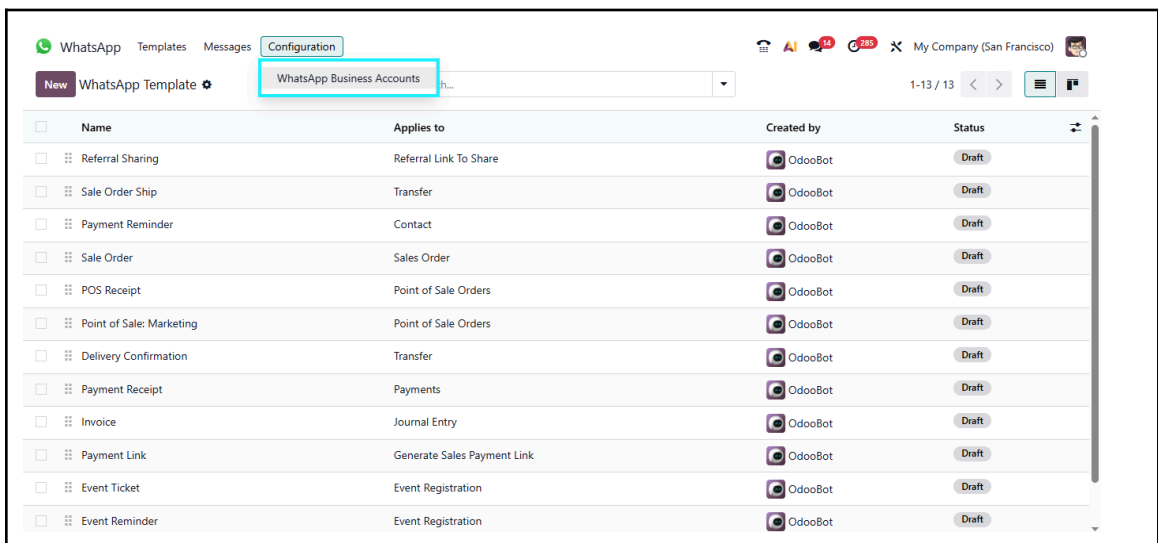


Langkah 4: Konfigurasi di Odoo

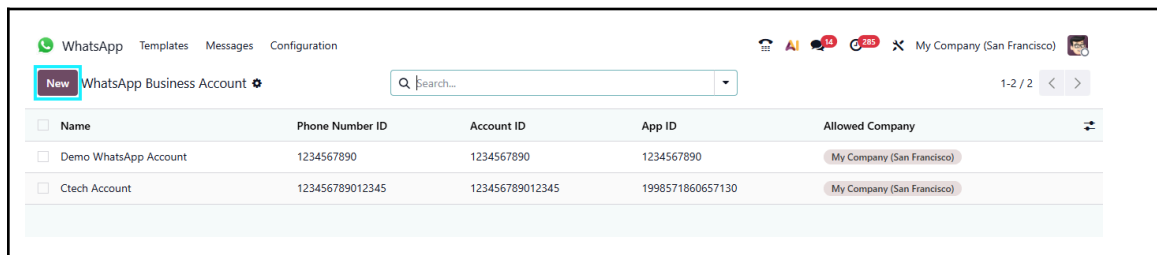
1. Masuk ke database Odoo dan klik modul Whatsapp. (Pasang terlebih dahulu jika belum menginstal modul Whatsapp dengan [tutorial berikut](#))



2. Pada Bar Navigasi, klik tombol Configuration dan Whatsapp Business Account.

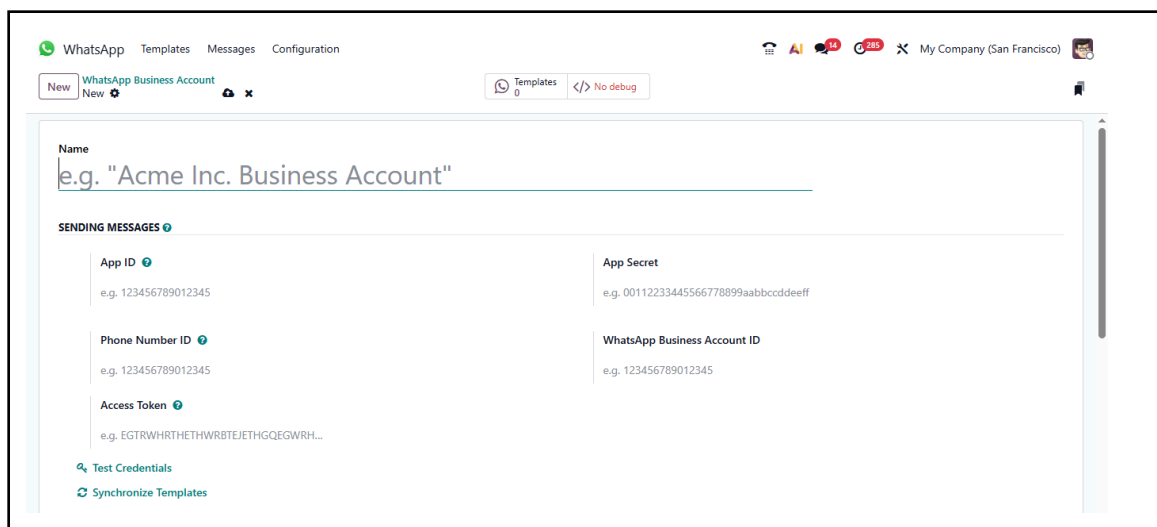


3. Klik tombol New untuk mulai membuat detail akun Whatsapp baru pada Odoo.



4. Masukkan dengan semua yang sudah didapatkan

- Application ID
- App Secret
- Phone Number ID
- Access Token
- WhatsApp Business Account ID



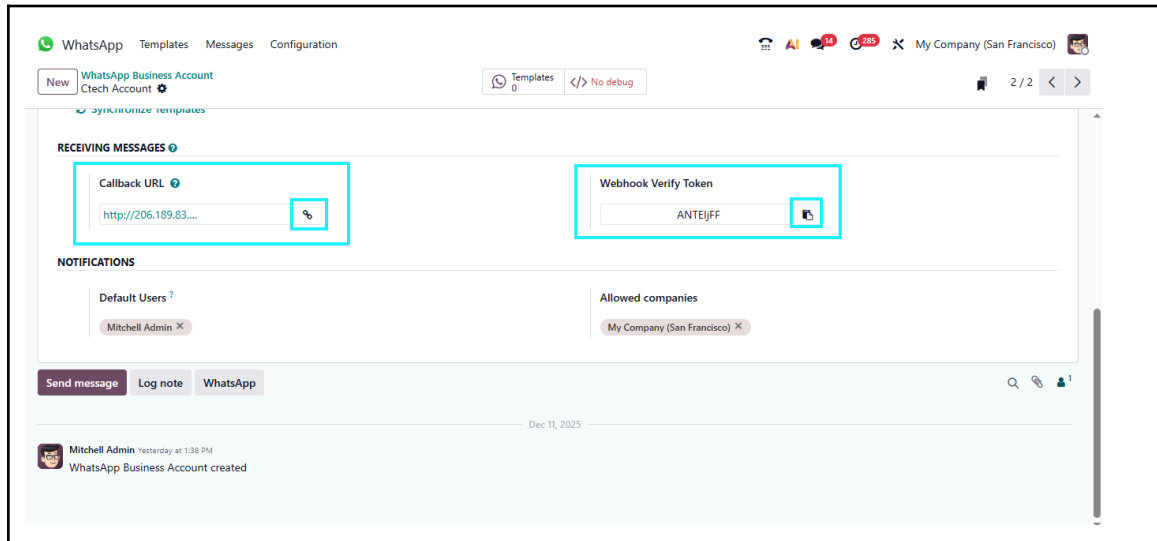
5. Klik "Save" lalu "Test Credentials".

Jika konfigurasi benar, kamu akan melihat pesan sukses. (Notifikasi berhasil terkoneksi)

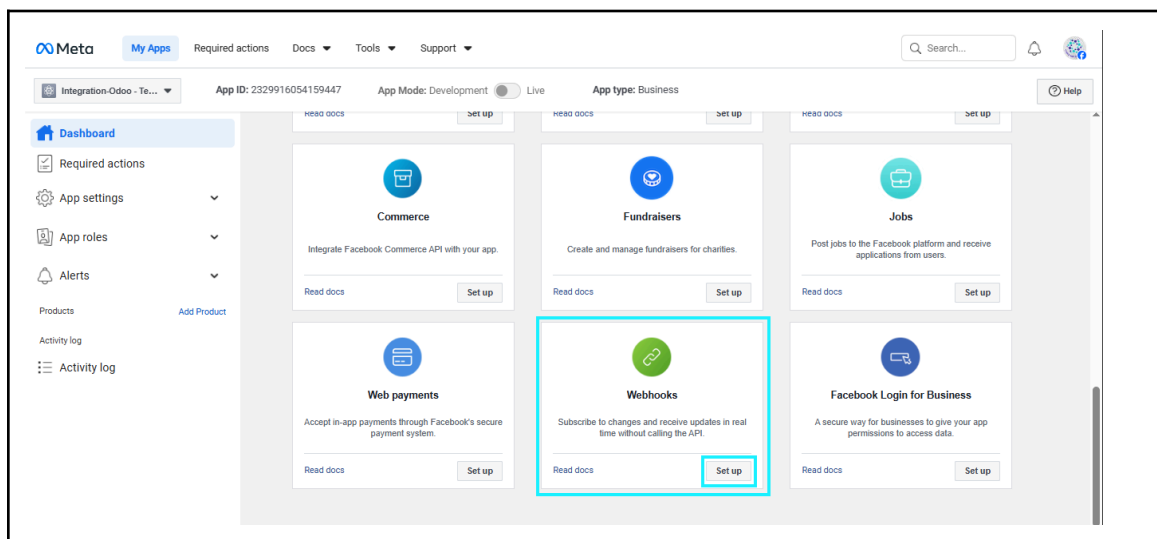
Langkah 5: Menyiapkan Webhook untuk Komunikasi Dua Arah

Agar Odoo bisa menerima balasan atau pesan masuk:

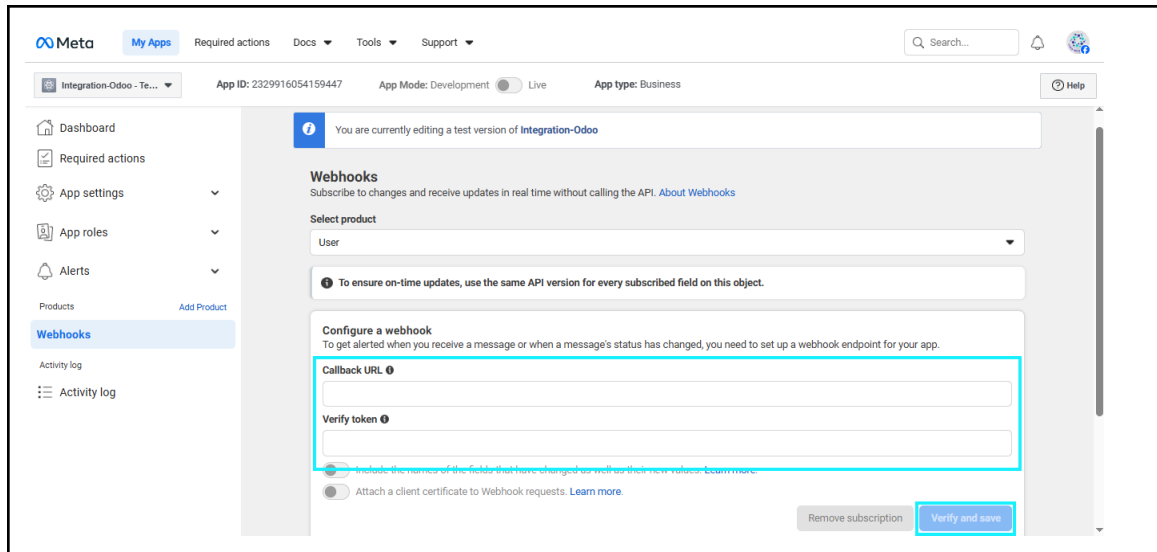
1. Pergi ke Navigasi bar “Configuration” dan klik “Whatsapp Business Account”. Scroll kebawah, salin “Webhook URL” dan “Verify Token”.



2. Kembali ke halaman Meta developer, pada Sidebar klik “Dashboard” dan cari kartu Webhooks. lalu lakukan setup dengan klik tombol Setup pada kartu Webhooks.

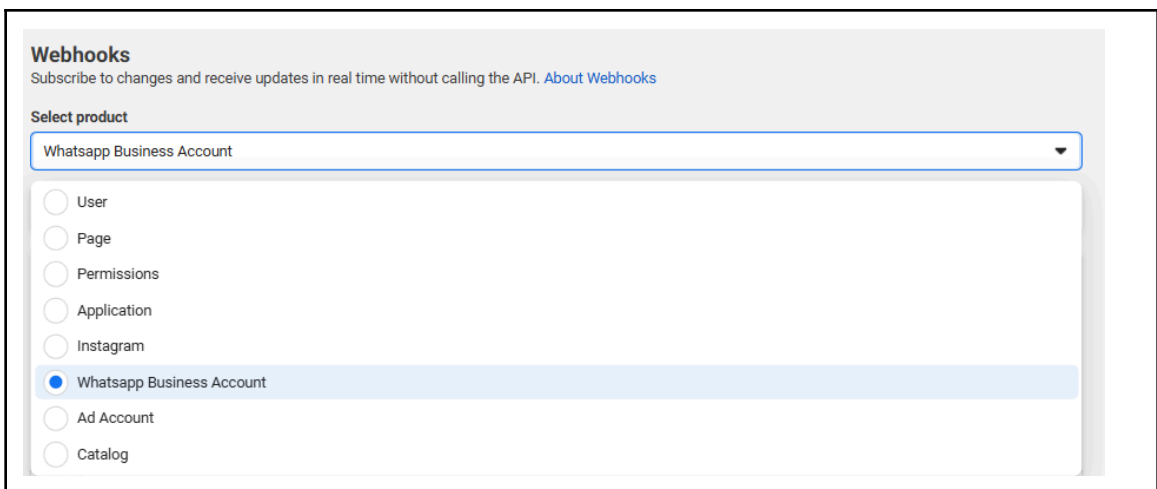


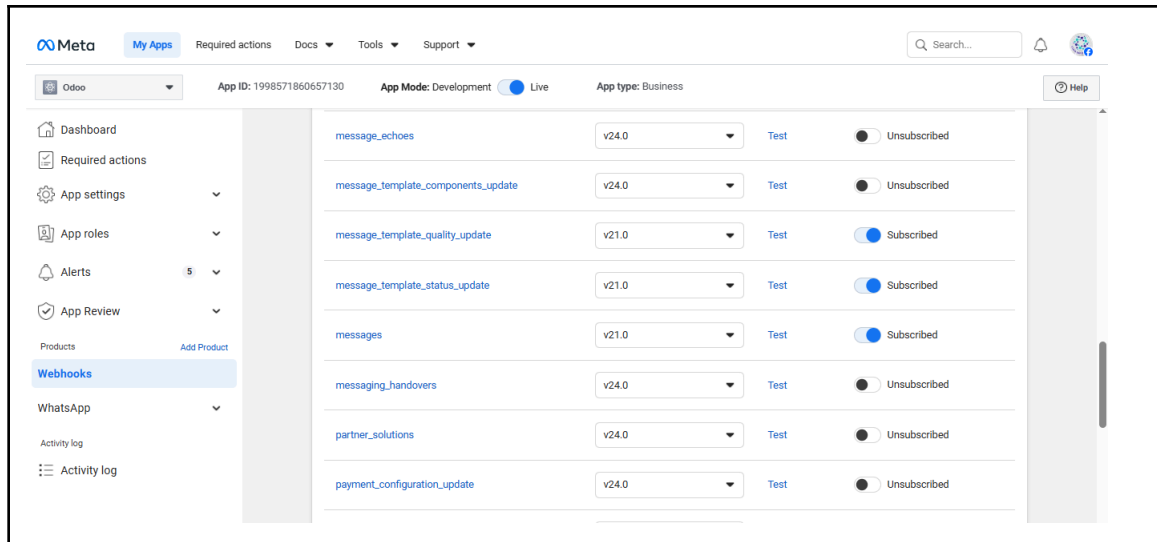
3. Di bagian “Webhooks”, tempel Webhook URL dan Verify Token dari Odoo.



4. Klik pada dropdown Select Product, lalu pilih produk Whatsapp Business Account. Lalu aktifkan / subscribe pada :

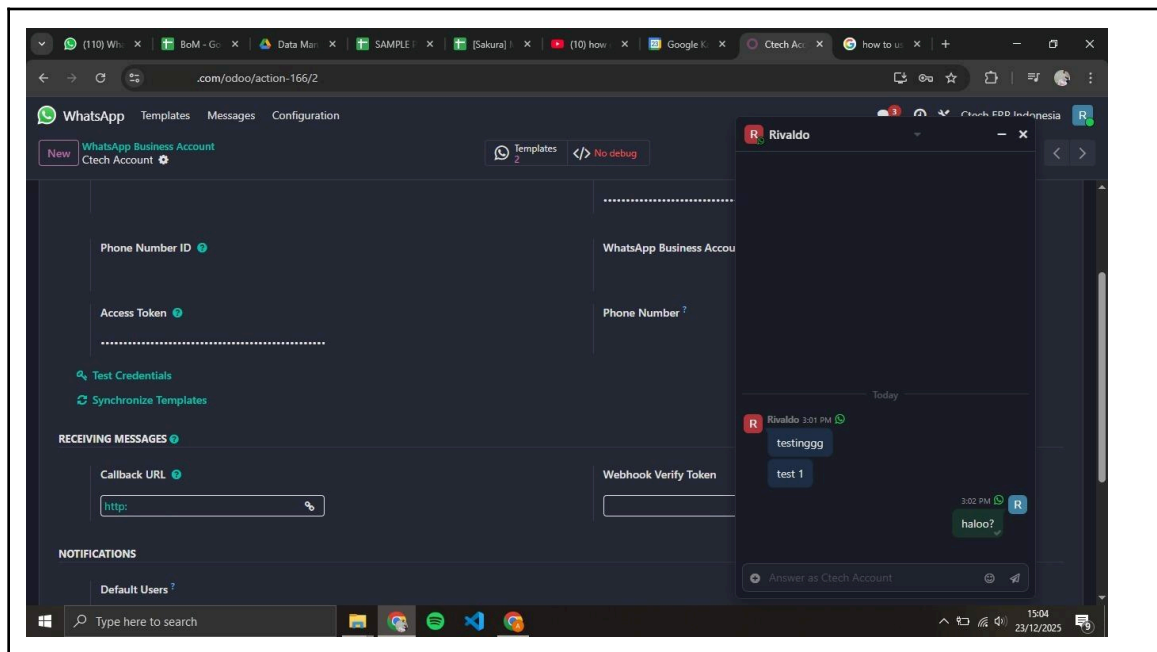
- messages
- Message_status_updates
- account_update
- template_update





Dengan ini Odoo bisa menerima dan memproses pesan masuk dari pelanggan.

- Setelahnya, dari sisi Client di whatsapp, coba untuk melakukan chatting terhadap nomor telepon yang digunakan Odoo (Nomor yang di daftarkan pada Meta Apps)



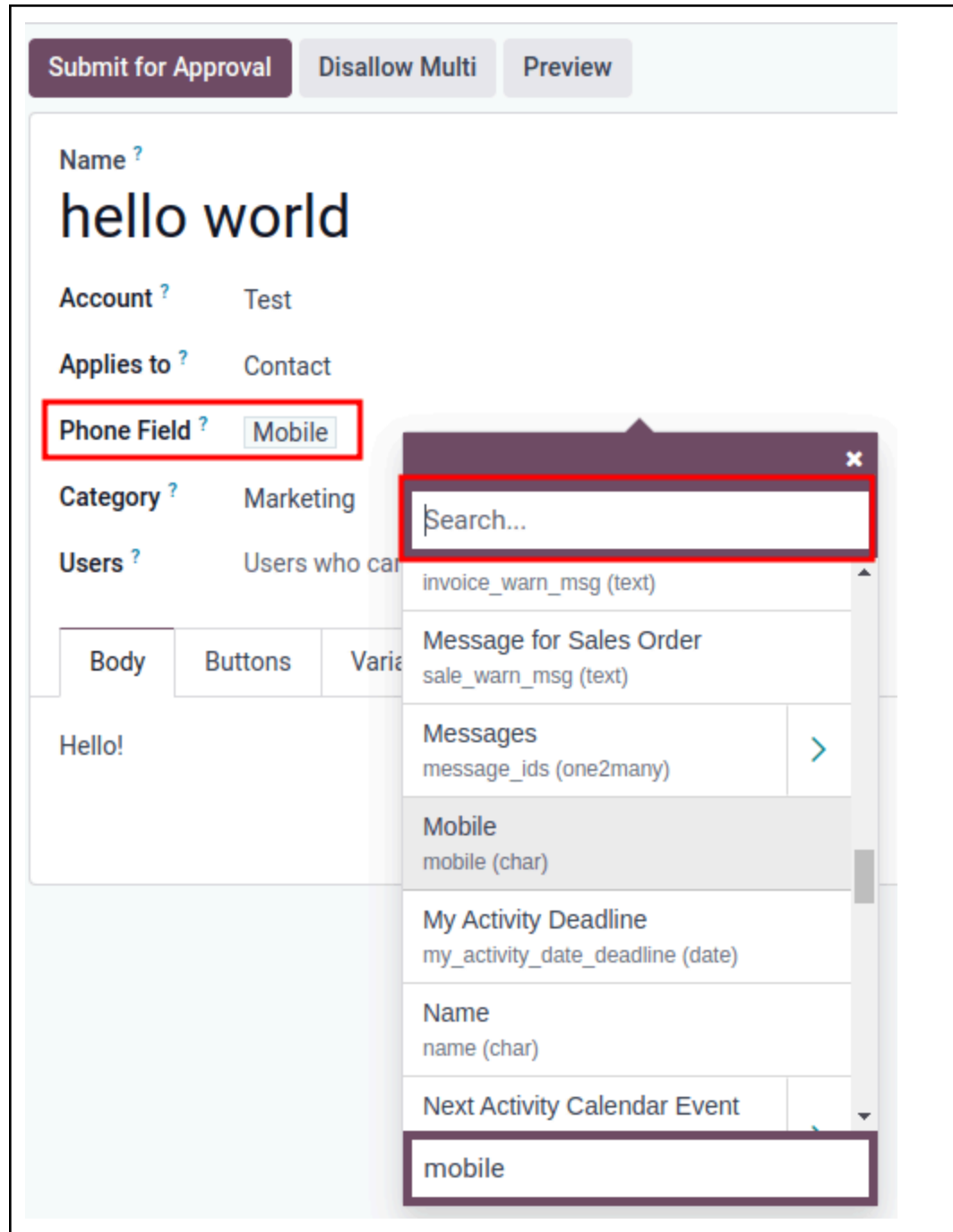
Langkah 6: Membuat Template WhatsApp di Odoo

Template diperlukan untuk mengirim pesan otomatis atau pesan bisnis ke pelanggan. Berikut cara membuatnya:

1. Di modul WhatsApp di Odoo, buka menu “Templates”.
2. Klik “Create” untuk membuat template baru.

Isi konfigurasi template:

- Nama template
- Kategori (misalnya Utility, Marketing, dsb.)
- Bahasa template
- Jenis header: bisa Text, Image, Document, atau Video
- Body pesan: isi teks dengan variabel placeholder jika perlu (misal {{1}}, {{2}}, dst.)



Sumber: [Odoo Documents](#)

3. Setelah selesai, kirim template untuk persetujuan (Submit for Approval).

Perlu diingat: semua template harus disetujui dulu oleh Meta sebelum bisa digunakan. Persetujuan biasanya memakan waktu hingga 24 jam. jadi tidak instan bisa digunakan saat itu juga.

Pemecahan Masalah Umum

❖ Pesan Tidak Terkirim

Jika pesan gagal terkirim, periksa hal berikut:

- Pastikan token akses belum kadaluarsa (gunakan token permanen).
- Pastikan kamu belum melewati batas pengiriman harian (jika berlaku) atau batas API.
- Pastikan template sudah disetujui oleh Meta.

❖ Tidak Menerima Balasan dari Pelanggan

Jika kamu tidak menerima pesan pelanggan di Odoo:

- Pastikan konfigurasi webhook (URL & Verify Token) sudah benar.
- Pastikan field “messages” sudah disubscribe di konfigurasi Webhook di Meta.
- Pastikan izin aplikasi sudah lengkap di Meta.

Jika ada error, cek log pesan gagal di Odoo, biasanya ada kode error yang dapat membantu anda untuk menemukan penyebab masalah.

Panduan Integrasi Odoo - Watzap



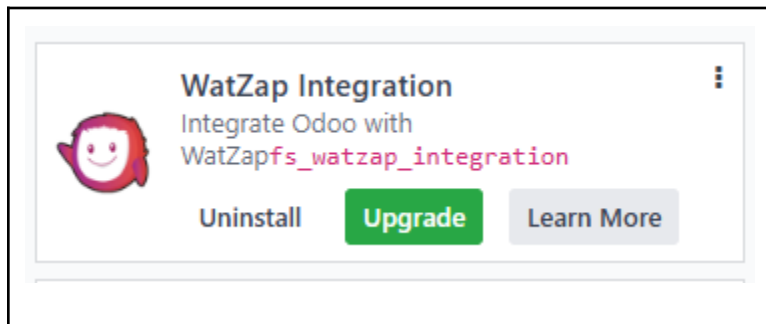
Panduan Integrasi Odoo - Watzap

Panduan ini dibuat diharapkan agar pembaca dapat memahami sekaligus mengetahui cara menggunakan Watzap untuk keperluan integrasi ke aplikasi yang sudah support integrasi dengan Watzap.

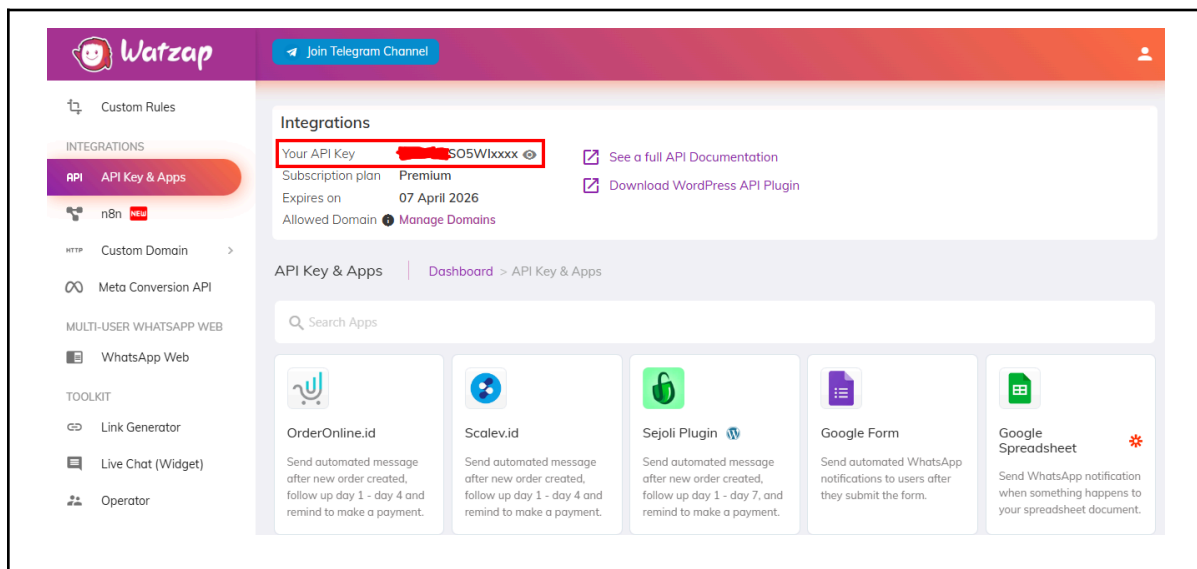
Created by: Billy & Rivaldo

Langkah Langkah Konfigurasi

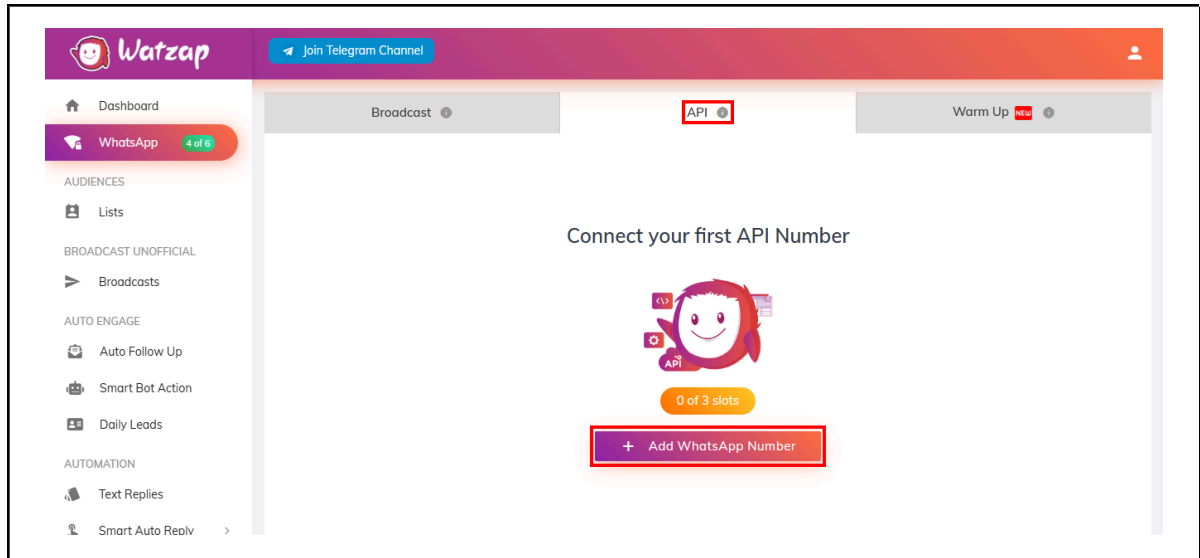
1. Pastikan pada odoo sudah terinstall modul watzap terlebih dahulu sebelum melakukan konfigurasi lebih lanjut. Jika modul sudah di push ke dalam server, instal pada database odoo yang ingin dilakukannya integrasi. [Tutorial instal ekstra modul bisa klik teks ini.](#)



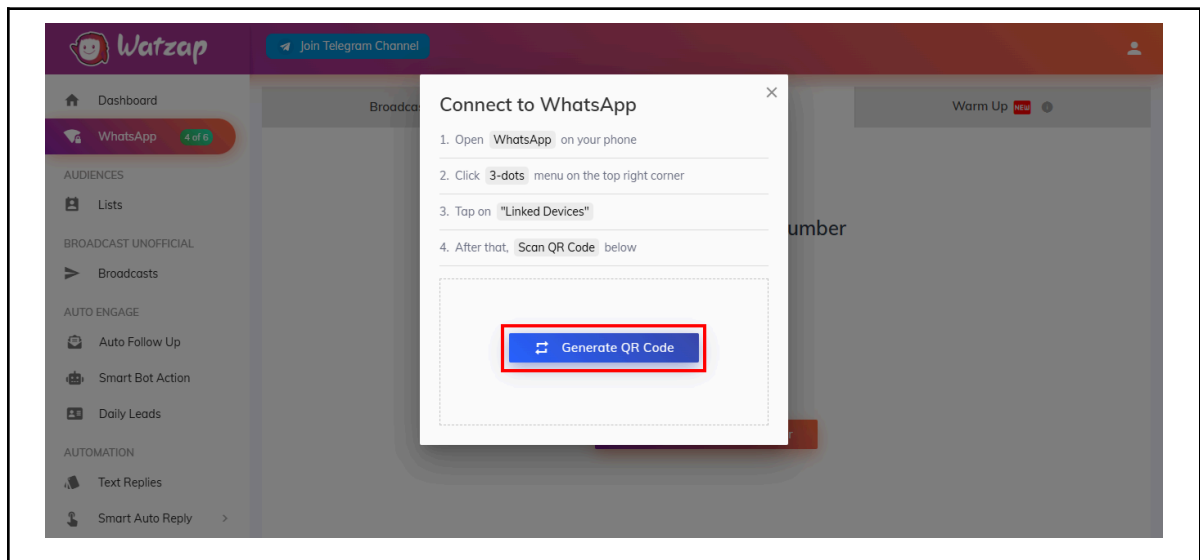
2. Lanjut, buka website [Watzap](#) lalu pada sidebar, klik menu **API Key & Apps** lalu pada **Your API Key**, klik icon clipboard untuk bisa copy kunci API Watzap. Simpan lalu, lanjut ke tahap selanjutnya.



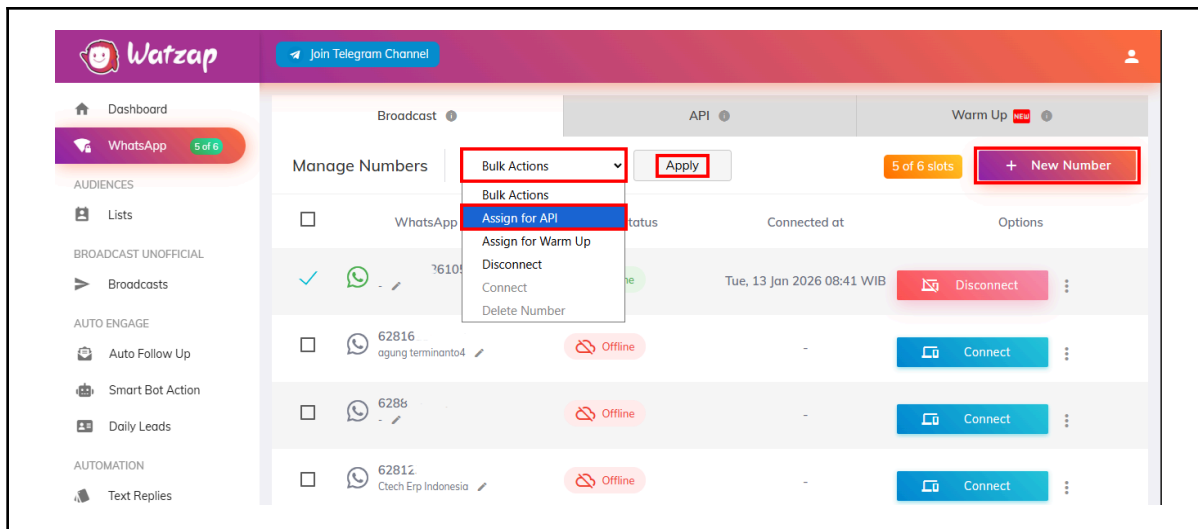
3. Pada sidebar, klik menu **Whatsapp** dan pilih tab **API**. Setelahnya, klik tombol **Add Whatsapp Number** untuk menambahkan nomor Whatsapp pada Watzap.



4. Buka Whatsapp pada hp, lalu buka menu tautkan perangkat (atau dalam bahasa inggrisnya **Linked Devices**) dan pada Watzap klik **Generate QR Code** lalu scan qr tersebut.

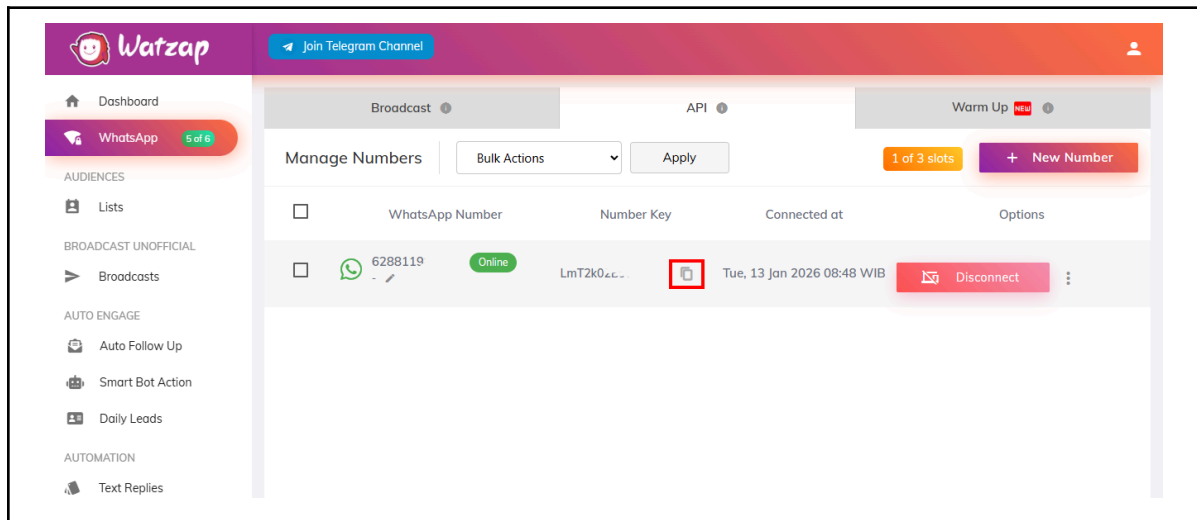


Sekedar tips...



*Jika pada saat **Generate QR Code** tidak kunjung muncul/ada kode error, bisa tambahkan nomor pada tab **Broadcast** lalu select number dan pada Action pilih **Assign for API**. Setelahnya klik **Apply**. Nanti akan diminta untuk scan QR ulang. Ikuti saja langkah-langkahnya yang diberikan pada popup di watzap.

5. Buka kembali tab **API**, lalu klik icon clipboard untuk copy **Number Key**. Simpan lalu lanjut ke konfigurasi di Odoo



6. Buka Odoo, pada menu kanban, klik modul **Watzap**, pada navigasi klik menu **Numbers** dan klik tombol **New** untuk mulai menambahkan Number Key Watzap dan jangan lupa klik tombol **Save**. Pada bagian **Name** itu tidak harus disamakan dengan Whatsapp. Itu hanya untuk inisial di Odoo saja. Jadi, menyesuaikan saja.

The image shows two screenshots of the Odoo 'Numbers' interface. The top screenshot shows the 'New' button highlighted in red. The bottom screenshot shows the 'Save' button highlighted in red, and the 'Name' field containing 'Ctech Assistant' and the 'Number Key' field containing 'LmT2k02L', both highlighted with red boxes.

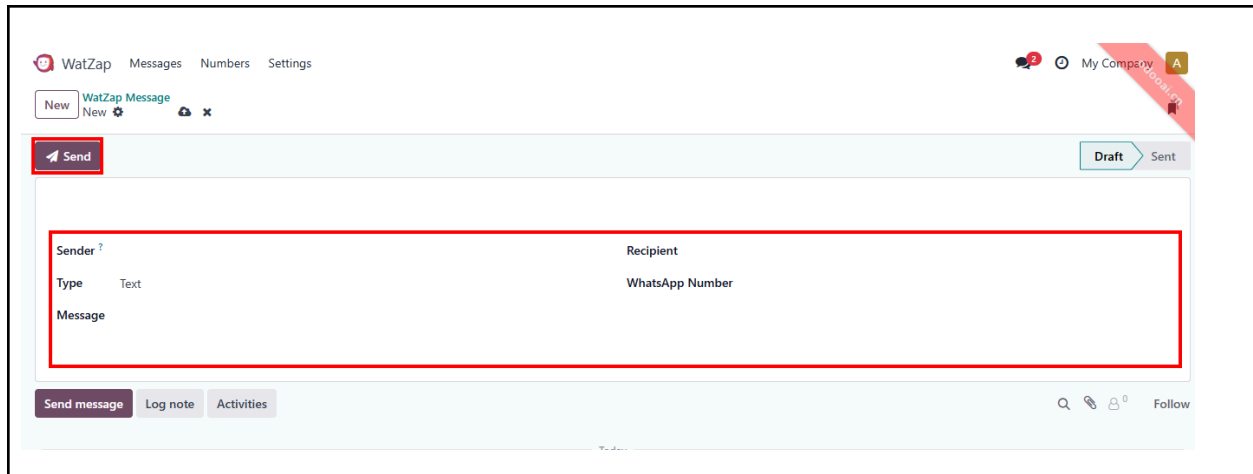
7. Selanjutnya, pada navigasi, klik menu **Settings**. Selanjutnya pada WatZap API Key, isikan API Key yang sudah di simpan sebelumnya. Pada bagian **Default Watzap Sender** klik inisial number sebelumnya. Sebagai contoh, saya membuat inisial **Ctech Assistant** dan terakhir klik tombol **Save**.

The image shows a screenshot of the Odoo 'WatZap Settings' interface. The 'Save' button is highlighted in red. The 'WatZap API Key' field contains 'XILBFE:..'. The 'Default WatZap Sender' dropdown menu is open, showing 'Ctech Assistant' highlighted with a red box.

Langkah Langkah Konfigurasi

Setelah melakukan konfigurasi, kita perlu melakukan testing/mencoba langsung untuk mengirim sebuah pesan.

Pada navigasi klik menu **Messages** dan klik tombol **New**.

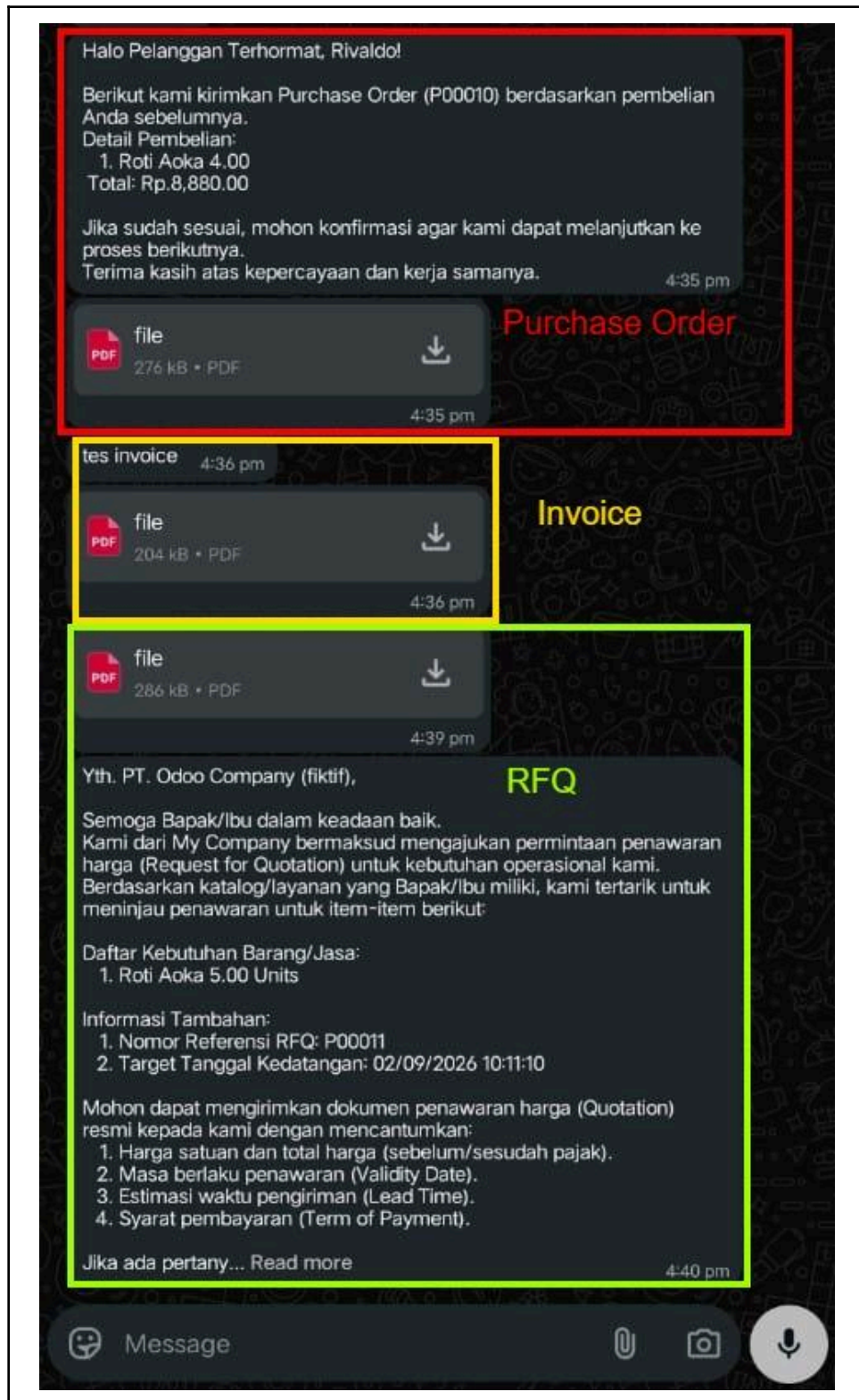


Setelahnya, kita perlu meng-input data yang kita punya untuk mulai mengirim pesan.

- Sender: Inisial Number yang sudah di input pada menu **Numbers**.
- Type: Pada type, kita bisa memilih untuk mengirim pesan dengan tambahan attachment. Teks, Dokumen, ataupun Gambar.
- Message: Disini kita dapat menambahkan isi pesan yang ingin disampaikan.
- Recipient: Sesuai namanya, jika pada Odoo sudah memiliki info kontak customer/seseorang yang ingin dikirim pesan, klik nama customer tsb.
- Whatsapp Number: Masukkan nomor whatsapp yang ingin dikirim pesan.

Klik **Send** untuk mulai mengirim pesan Whatsapp. Tunggu sekitar 2 detik sampai pesan terkirim.

Contoh Pesan Yang Dikirim Menggunakan Modul Kustom Watzap Ini



Analisis Komparasi Teknis Whatsapp Odoo



Analisis Komparasi Teknis Terhadap Integrasi Whatsapp Odoo

Odoo Standard Integration vs. Watzap Custom Module

Created by: Billy & Rivaldo

A. Pendahuluan

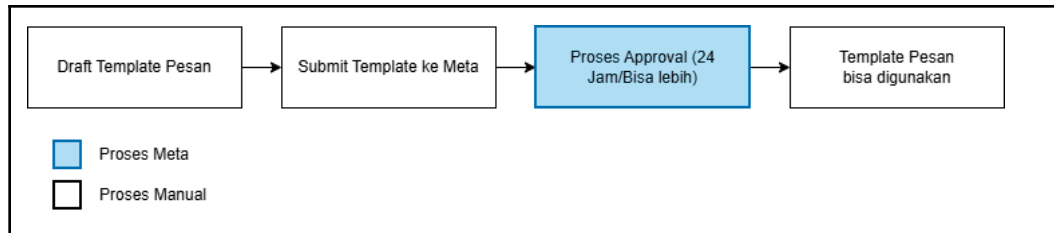
Panduan ini disusun untuk memberikan gambaran objektif mengenai perbedaan efisiensi antara integrasi WhatsApp standar (Meta) dengan *Module Custom* Watzap. Tujuannya agar pembaca dapat menentukan solusi yang paling optimal dari sisi operasional dan kecepatan implementasi.

B. Perbandingan Integrasi

1. Module Odoo Default (Whatsapp Odoo 19)

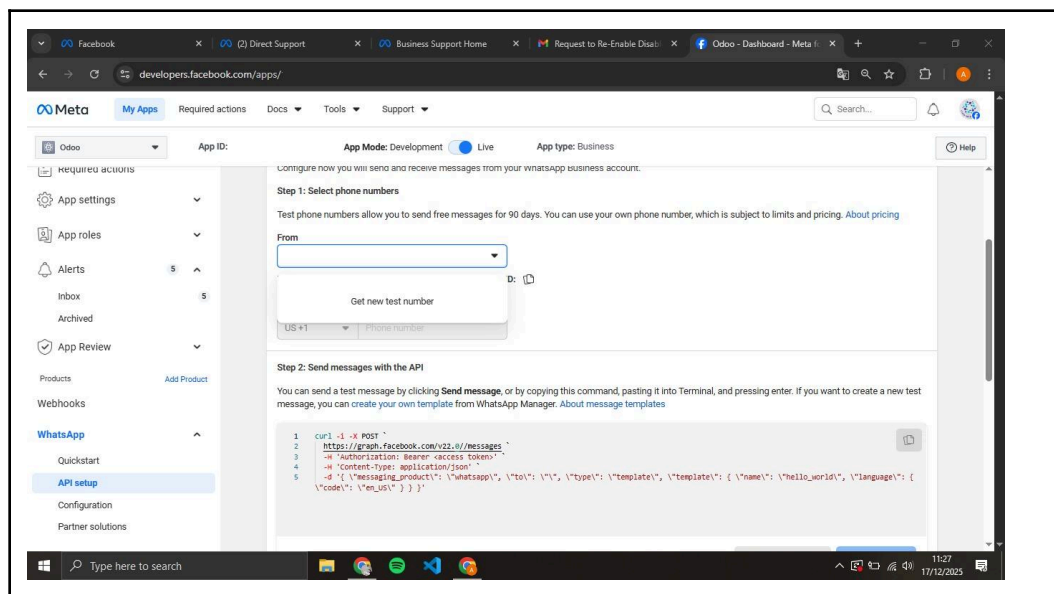
Pendekatan pertama adalah menggunakan fitur bawaan Odoo yang menginduk langsung pada ekosistem Meta. Metode ini menawarkan konektivitas resmi, namun menuntut tingkat pemahaman teknis yang lebih tinggi dalam proses implementasinya.

- **Arsitektur dan Platform:** Integrasi dilakukan secara langsung melalui *Meta for Developers*. Pendekatan ini cocok untuk skala *enterprise* yang membutuhkan jalur komunikasi resmi dari WhatsApp.
- **Proses Konfigurasi:** Alur pengaturan (*setup*) terbilang berlapis dan sangat teknis. Pengguna diwajibkan untuk membangun aplikasi WhatsApp di dalam dasbor Meta, mengatur *Webhook* untuk memastikan komunikasi dua arah berjalan lancar, dan mengkonfigurasikannya ke dalam sistem Odoo.
- **Kebijakan Pesan:** Rumit. Kalau membuat sebuah *template* pesan untuk digunakan implementasi nanti, *template* wajib disetujui (*Approval*) oleh pihak Meta itu sendiri sebelum akhirnya pengguna dapat menggunakan *template* pesan.



Gambar: Flow Ketika Mencoba Menggunakan Template Meta

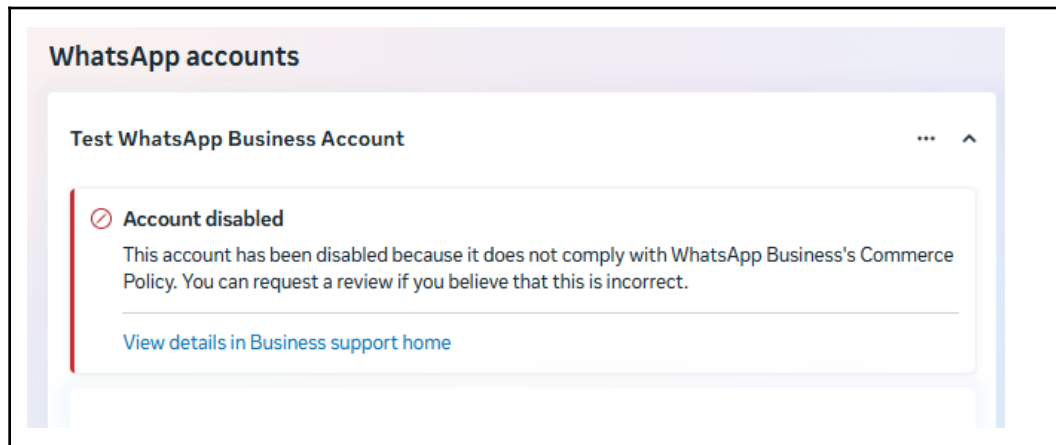
- **Autentikasi dan Kredensial:** Sistem membutuhkan input data teknis yang cukup kompleks sebagai penghubung. Beberapa variabel yang wajib dimasukkan meliputi *Application ID*, *App Secret*, *Phone Number ID*, *Access Token*, serta *WhatsApp Business Account ID*.



Gambar: Mengkonfigurasi Data Teknis Meta

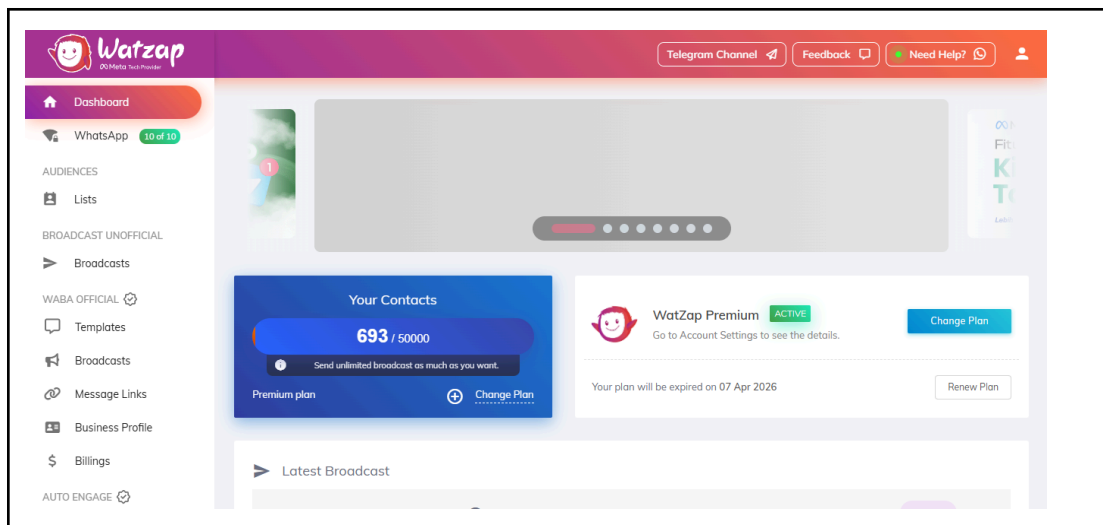
- **Kebijakan Pengiriman Pesan:** Terdapat regulasi yang ketat terkait pengiriman pesan. Pengguna diwajibkan menggunakan *Template* WhatsApp untuk memulai percakapan (*outbound message*). Setiap *template* yang dibuat harus melewati proses moderasi dan persetujuan (*Submit for Approval*) dari pihak Meta, yang umumnya memakan waktu hingga 24 jam sebelum dapat digunakan.
- **Akun Whatsapp Terkena Suspend:** Masih sangat mungkin apabila akun whatsapp masih tergolong baru, *Whatsapp Business Messages* dapat terkena suspend karena di anggap Spam ketika pertama kali mengirim pesan kepada seseorang yang tidak pernah

dihubungi sebelumnya. Jika sudah terkena suspend, agak sulit untuk dapat melakukan banding ke Tim Whatsapp.



Gambar: Akun WABA Terkena Suspend

2. Modul Kustom (Watzap.id - pihak ketiga)

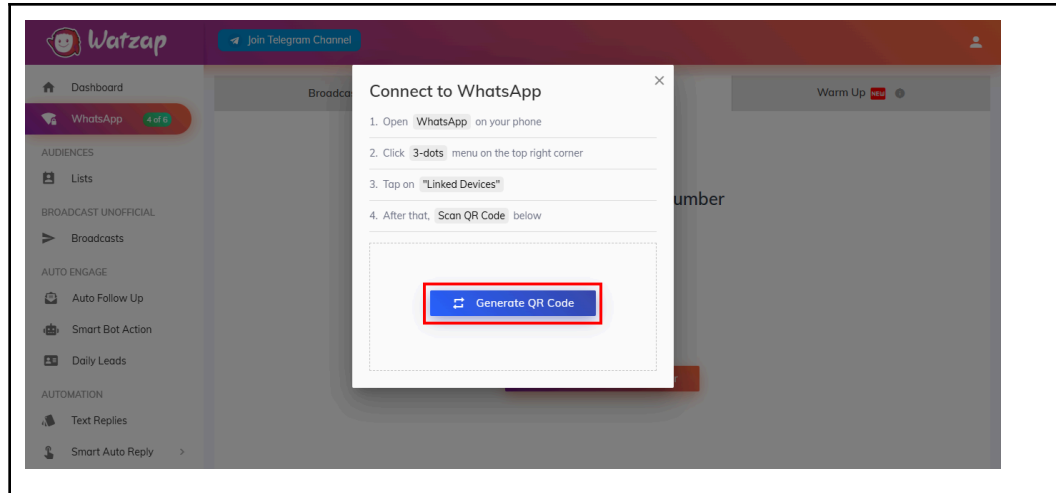


Gambar: Tampilan Dasbor Watzap.id

Pendekatan kedua berfokus pada efisiensi dan kemudahan operasional menggunakan modul pihak ketiga, yaitu platform Watzap (app.watzap.id). Metode ini secara signifikan memangkas birokrasi teknis yang ada pada integrasi resmi.

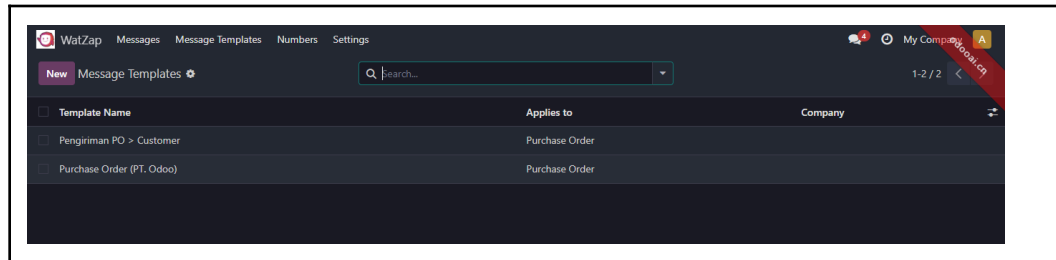
- **Proses Konfigurasi & Autentikasi:** Implementasi dirancang agar sangat praktis dan intuitif. Alih-alih menggunakan *webhook* yang rumit, pengguna hanya perlu menautkan nomor WhatsApp melalui pemindaian *QR Code* pada fitur "Tautkan Perangkat" (*Linked Devices*), persis seperti mekanisme WhatsApp Web. Parameter kredensial juga

disederhanakan; pengguna hanya perlu menyalin *API Key* dan *Number Key* dari dasbor Watzap ke dalam menu Odoo.



Gambar: Melakukan *Linked Devices* ke [Watzap.id](https://watzap.id)

- **Kebijakan Pengiriman Pesan:** Pendekatan ini menawarkan fleksibilitas penuh tanpa hambatan birokrasi *template*. Pengguna dapat langsung mengetik dan mengirimkan pesan. Baik berupa teks, dokumen, maupun gambar secara *real-time* melalui menu *Messages* tanpa perlu menunggu persetujuan (*approval*) dari pihak manapun.



Gambar: *Messages Template* dengan *Module Watzap.id*

C. Tabel Ringkasan Komparasi

Fitur / Aspek	Odoo Default (Meta Apps)	Module Custom (Watzap)
Segi Koneksi	Webhooks Meta Developers	Scan QR Code (WhatsApp Web)
Proses Setup	Rumit (Konfigurasi Webhook & App)	Instan (Tautkan Perangkat)
Kredensial	App ID, Secret Token, Phone ID	API Key & Number Key

Aturan Pesan	Wajib <i>Template & Approval</i> (24 Jam)	Bebas (Tanpa Template/Approval)
Biaya & Prosedur	Mengikuti kebijakan resmi Meta	Berlangganan via Dashboard Watzap
Fleksibilitas	Terbatas pada aturan Business API	Sangat Tinggi (Seperti WA Reguler)

D. Kesimpulan

Berdasarkan analisis di atas, dapat ditarik kesimpulan sebagai berikut:

1. **Efisiensi Operasional:** *Module* Watzap unggul mutlak dalam kecepatan implementasi. Proses penautan perangkat hanya melalui *QR Code* memangkas waktu tunggu persetujuan *template* dari Meta yang biasanya memakan waktu hingga 24 jam atau malah bisa lebih dari itu.
2. **Kemudahan Penggunaan:** Integrasi *Module* Whatsapp Odoo 19 memerlukan keahlian teknis khusus untuk mengelola kredensial Meta. Sebaliknya, Watzap menawarkan antarmuka yang jauh lebih ramah pengguna (*user-friendly*) karena hanya membutuhkan dua kunci utama (*API & Number Key*).
3. **Rekomendasi:** Gunakan *Module* Whatsapp yang ada pada versi terbaru Odoo 19 jika perusahaan memerlukan status *Official Business Account* (Centang Biru) dan siap dengan birokrasi secara teknis Meta.
 - Gunakan *Module* Watzap.id ini jika prioritas utama Anda adalah kecepatan pengiriman pesan tanpa batas, kemudahan *setup*, dan fleksibilitas konten pesan (teks/gambar/dokumen) secara langsung.