

## Perancangan Dan Implementasi *Bot Telegram* Untuk Pemetaan Lokasi ODP Dan Akses Informasi Di Telkom Akses Kota Samarinda

Alvin Hakim

Universitas Pembangunan Nasional “Veteran” Jawa Timur

E-mail : [20032010168@student.upnjatim.ac.id](mailto:20032010168@student.upnjatim.ac.id)

Tranggono Tranggono

Universitas Pembangunan Nasional “Veteran” Jawa Timur

E-mail : [tranggono.ti@upnjatim.ac.id](mailto:tranggono.ti@upnjatim.ac.id)

**Abstract.** Facing the dynamic landscape of the telecommunications industry, this research delves into key challenges in the operational efficiency of network management. The approach involves designing and implementing an innovative Telegram bot, specifically crafted to map the locations of Optical Distribution Points (ODP). This bot enables technicians to easily access information by entering the ODP name through the Telegram platform. The research methodology employs a descriptive method and a development method using the waterfall SDLC (Software Development Life Cycle) concept, suitable for small-scale projects. Unified Modeling Language (UML), including Use Case Diagrams and Activity Diagrams, aids in the system design. The author conducted an in-depth identification of field technicians' requirements, storing ODP data in Google sheets, including ODP name, geographical coordinates, address, partner, (GPON), and status. The Telegram bot implementation displays location information linked directly to Google Maps, providing essential details such as address, partner, type (GPON), and ODP status. This Telegram bot implementation demonstrates a significant positive impact, contributing to the effectiveness and empowerment of field technicians in managing ODP data. Discussions involving bot success evaluation and potential work effectiveness underscore positive outcomes. The research anticipates that all technicians at PT Telkom Akses Samarinda can utilize this bot to enhance their operational workflows.

**Keywords:** Optical Distribution Point (ODP), Telecommunications industry, Telegram Bot.

**Abstrak.** Dalam menghadapi dinamika industri telekomunikasi, penelitian ini fokus pada efisiensi operasional manajemen jaringan. Dengan merancang dan mengimplementasikan *bot Telegram* inovatif, khusus untuk memetakan lokasi *Optical Distribution Point* (ODP), para teknisi dapat dengan mudah mengakses informasi melalui platform *Telegram*. Metode pengembangan menggunakan konsep *waterfall* SDLC, cocok untuk proyek skala kecil, didukung oleh *Unified Modeling Language* (UML) seperti *Use Case Diagram* dan *Activity Diagram*. Penelitian melibatkan identifikasi kebutuhan teknisi lapangan, dengan data ODP disimpan di *Google Sheet*, mencakup nama ODP, koordinat, alamat, mitra, (GPON), dan status. Implementasi *bot Telegram* menampilkan lokasi terhubung ke *Google Maps*, serta detail seperti alamat, mitra, jenis (GPON), dan status ODP. Hasilnya menunjukkan dampak positif, meningkatkan efektivitas dan kemandirian teknisi lapangan dalam mengelola informasi data ODP. Keberhasilan *bot* ini diharapkan memberikan manfaat kepada semua teknisi di PT Telkom Akses Samarinda untuk meningkatkan operasional mereka.

**Kata kunci:** *Bot Telegram*, Industri Telekomunikasi, *Optical Distribution Point* (ODP)

### PENDAHULUAN

Dalam menghadapi pesatnya pertumbuhan kebutuhan internet, PT. Telkom Akses memperluas infrastruktur jaringan *wifi* dengan membangun *Optical Distribution Point* (ODP) yang tersebar di seluruh kota Samarinda. Meskipun langkah ini diambil untuk mengakomodasi meningkatnya pelanggan dari layanan Indihome, kompleksitas pemeliharaan dan manajemen ODP menjadi tantangan yang perlu diatasi untuk memastikan kelangsungan operasional dan daya saing industri telekomunikasi. Keberlanjutan dan efisiensi operasional dianggap sebagai

pilar utama dalam menjawab ekspektasi konsumen yang terus meningkat. Oleh karena itu, peran informasi dalam proses perbaikan ODP menjadi sangat krusial. Teknisi membutuhkan akses cepat terhadap berbagai informasi terkait ODP, seperti alamat lengkap, titik koordinat, status, mitra, dan GPON, agar dapat merespons dan memperbaiki gangguan dengan efektif. Pada kenyataannya, banyaknya permintaan informasi mengenai ODP kepada tim kantor dapat menghambat proses pemberian informasi tersebut. Begitu pula, teknisi harus menunggu informasi dari tim kantor terkait ODP yang akan diperbaiki. Keterlambatan ini dapat memperlambat respon teknisi, mengakibatkan peningkatan downtime, dan secara keseluruhan mengganggu efisiensi operasional

Dalam menanggapi kompleksitas tersebut, penelitian ini mengusulkan solusi inovatif melalui perancangan dan implementasi *bot Telegram*. *Telegram* merupakan satu-satunya aplikasi pesan singkat yang menyediakan api bagi pengguna untuk dapat membuat *bot* yang bisa dimanfaatkan untuk sistem informasi (Prastowo dkk., 2019). *Bot* merupakan aplikasi pihak ketiga yang dapat dijalankan di dalam *Telegram*. Pengguna dapat mengirim pesan, perintah, dan inline request. Kita dapat mengontrol *bot* menggunakan HTTPS ke API *Telegram* (Mulyanto, 2020). *Bot Telegram* diharapkan dapat memfasilitasi komunikasi yang lebih efektif antara tim kantor dan teknisi. Melalui aplikasi ini, teknisi hanya perlu menginput nama ODP yang dimaksud, dan *bot Telegram* akan memberikan informasi yang diperlukan, seperti alamat lengkap, titik koordinat, status, mitra, dan GPON. Penggunaan *bot Telegram* dipilih karena kecepatan dan kemudahan komunikasinya, serta kemampuannya untuk menyediakan informasi secara instan. Hal ini dianggap sebagai langkah proaktif untuk meningkatkan waktu tanggap teknisi dan mengoptimalkan efisiensi operasional secara keseluruhan. Pendekatan metodologi yang digunakan adalah deskriptif dengan konsep *waterfall SDLC*. Metode ini dipilih untuk memastikan proses pengembangan perangkat lunak yang terstruktur dan dapat dikelola dengan baik. Selain itu, Unified Modeling Language (UML) digunakan untuk mendukung pemodelan, dengan Use Case Diagram dan *Activity Diagram* membantu menggambarkan interaksi antara pengguna dan sistem.

Penelitian ini tidak hanya bertujuan untuk menangani masalah saat ini tetapi juga berfokus pada keberlanjutan manajemen jaringan di PT. Telkom Akses Kota Samarinda. Dengan memberikan solusi yang responsif dan efektif, penelitian ini diharapkan dapat memberikan dampak positif dalam mendukung perkembangan industri telekomunikasi secara keseluruhan, menciptakan efisiensi operasional, dan meningkatkan kepuasan pelanggan.

## KAJIAN TEORITIS

### *Telegram*

Teknologi komunikasi telah banyak berubah selama 20 tahun terakhir. Dari dulu yang awalnya dihandphone hanya ada sms dan telepon sekarang sudah ada aplikasi *instant messaging* yang sudah mengakomodir keduanya bahkan bisa untuk *video call* dan masih banyak fitur lain. Saat ini telah banyak aplikasi pesan instan (IM) seperti *Whatsapp*, *Line*, *Snapchat*, *Facebook Messenger* dan *Telegram* (Gregorio, et.al., 2017). *Telegram* merupakan satu-satunya aplikasi pesan singkat yang menyediakan api bagi pengguna untuk dapat membuat *bot* yang bisa dimanfaatkan untuk sistem informasi (Prastowo, et.al., 2019).

### *Bot Telegram*

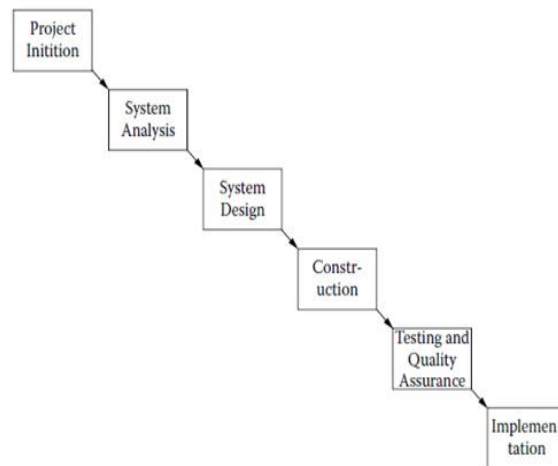
*Bot* merupakan aplikasi pihak ketiga yang dapat dijalankan di dalam *Telegram*. Pengguna dapat mengirim pesan, perintah, dan inline request. Kita dapat mengontrol *bot* menggunakan HTTPS ke API *Telegram*.

*Bot* atau *robot* biasa digunakan untuk kegiatan otomatisasi terhadap sebuah kegiatan yang diulang-ulang, serta dapat digunakan sebagai alat pengawasan/monitoring yang dilakukan oleh pihak admin (Atmojo, 2018).

*Bot Telegram* menjadi *bot* yang saat ini populer digunakan oleh banyak orang di berbagai instansi untuk mendukung kegiatan yang dilakukannya. Di Telkom Akses sendiri penggunaan *bot* sudah banyak diterapkan kepada para pegawai perusahaan, *bot Telegram* sangat membantu para pegawai dalam membantu menjalankan pekerjaannya

## METODE PENELITIAN

Untuk metode pengembangan yang diterapkan dalam perancangan *bot* menggunakan pendekatan *waterfall* SDLC. Pendekatan *waterfall* adalah model yang sering digunakan oleh Software Engineering dimana pengembangannya secara berurutan dan mengalir ke bawah seperti air terjun (Zulfitra dan Ayuningtyas, 2023) . Model *waterfall*, atau yang umumnya dikenal sebagai model air terjun, adalah suatu pendekatan dalam pengembangan perangkat lunak yang membagi kegiatan dasar, seperti pengembangan, validasi spesifikasi, dan evolusi, ke dalam serangkaian fase yang diproses secara berurutan. Model ini juga dikenal dengan sebutan *waterfall* atau Siklus Hidup Klasik. Tahapan dalam model proses ini terdiri dari enam langkah, sebagaimana tergambar pada Gambar



**Gambar 1.** Alur Pengembangan Sistem dengan Pendekatan *Waterfall* SDLC

1. Analisa sistem

Analisa kebutuhan perangkat lunak Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain aplikasi

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean.

3. Implementasi Sistem

Proses implementasi melibatkan konversi desain aplikasi menjadi kode program, membentuk modul-modul, dan menggabungkannya pada tahap berikutnya. Langkah ini bertujuan untuk menyelesaikan desain sistem yang telah disetujui, menginstalasi, dan memulai penggunaan sistem baru atau yang telah diperbaiki. Dalam tahap implementasi, tujuan utamanya adalah menyelesaikan desain sistem yang telah mendapatkan persetujuan, menguji dan mendokumentasikan program-program serta prosedur sistem yang diperlukan, memastikan bahwa pengguna terlibat dapat mengoperasikan sistem baru, dan memverifikasi bahwa migrasi dari sistem lama ke sistem baru dapat berjalan dengan baik dan tepat..

4. Pengujian (*testing*)

Setiap modul akan digabungkan dan diuji untuk menilai sejauh mana sistem dapat dioperasikan oleh pengguna. Metode pengujian yang digunakan adalah *Black Box Testing*, fokus pada evaluasi tampilan dan fitur sistem tanpa memerhatikan struktur internal. Kasus uji dibuat berdasarkan persyaratan dan spesifikasi, tidak memerlukan pemahaman mendalam

tentang kode aplikasi. Perancang tes mengevaluasi respons terhadap *input* valid dan tidak valid tanpa memahami komposisi internal item yang diuji.

#### 5. Verifikasi

Pada tahapan ini, pengguna (*user*) langsung melakukan uji coba pada sistem apakah sudah sesuai dengan standar. Tahapan ini bertujuan untuk memverifikasi *bot* yang sudah dibuat untuk selanjutnya agar bisa digunakan oleh Teknisi PT.Telkom Akses Samarinda.

#### 6. Pemeliharaan (*maintenance*)

Tahap pemeliharaan merupakan langkah terakhir setelah sistem menjalani pengujian, dan jika terdapat kesalahan pada tahap sebelumnya, perbaikan atau pemeliharaan akan dilakukan. Pemeliharaan disarankan dilakukan secara berkala, minimal dua kali pengecekan. Pengecekan pertama direkomendasikan dilakukan setelah satu bulan sistem diluncurkan, sedangkan pengecekan kedua dilakukan setelah enam bulan sistem berjalan. Tujuan dari pemeliharaan ini adalah untuk memastikan bahwa sistem tetap beroperasi sesuai dengan desain awal yang disepakati, sekaligus menjadi bahan evaluasi dan dasar pengembangan berkelanjutan. Proses ini memberikan kesempatan untuk perbaikan atau penyempurnaan ulang dari sistem yang telah dibuat (Zulfitra dan Ayuningtyas, 2023).

## **HASIL DAN PEMBAHASAN**

### **Teknik Pengumpulan Data**

Perancangan dan implementasi *bot* ODP kordinat menggunakan *Apps script* dilakukan dari bulan September 2023 hingga November 2023. Proses dimulai dengan tahap wawancara bersama tim kantor dan teknisi, diikuti oleh observasi, perancangan desain, hingga mencapai tahap pemeliharaan. Berikut adalah penjelasan secara lebih rinci.

### **Analisis Sistem**

Berdasarkan hasil wawancara dan observasi yang telah dilakukan, maka sistem informasi *bot* didapatkan hal-hal sebagaimana tabel berikut :

**Tabel 1. Analisis Sistem**

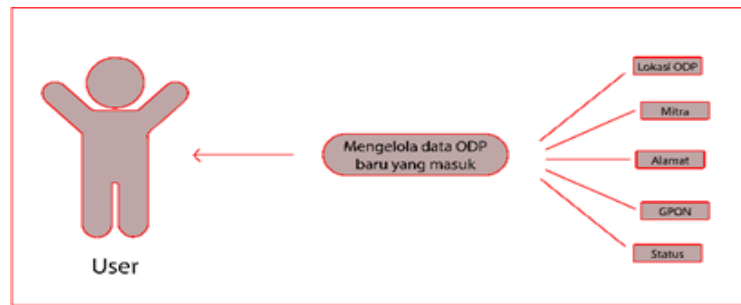
Faktor		Hasil Analisa
1.	Identifikasi Masalah	Saat ini belum ada sistem <i>bot</i> koordinat di PT. Telkom Akses Samarinda untuk menyampaikan data <i>informasi</i> ODP kepada para teknisi. Akibatnya, metode penyampaian <i>informasi</i> ODP masih dilakukan secara manual melalui tim teknisi menghubungi tim kantor dan belum menggunakan sistem otomatis. Hal ini dapat menghambat dan memperlambat para pekerja baik dari tim kantor maupun tim teknisi, sehingga efektivitas dan efisiensi kerja tidak dapat dicapai.
2.	Ruang Lingkup Sistem	Ruang lingkup penelitian sistem terfokus pada satu tempat saja yaitu laboratorium PT. Telkom Akses Samarinda
3.	Manfaat Sistem	<i>Bot</i> ini melayani sebagai solusi untuk meningkatkan responsivitas dan efisiensi kerja tim teknisi di lapangan. Dengan demikian, meminimalkan gangguan yang mungkin timbul akibat peningkatan permintaan <i>informasi</i> lokasi ODP.
4.	Kebutuhan Sistem	Kebutuhan sistem adalah segala sesuatu yang menunjang agar sistem dapat berjalan meliputi hal-hal apa saja yang berkaitan dan dibutuhkan diantaranya: <ul style="list-style-type: none"> <li>a. <i>Google sheets</i> berperan sebagai database untuk menyimpan data ODP berupa nama ODP, koordinat, alamat, mitra, GPON dan status ODP.</li> <li>b. <i>API bot Telegram</i> yang akan dipakai.</li> <li>c. <i>Script</i> dari <i>Apps script</i> dimana <i>script</i> tersebut langsung terhubung dengan <i>bot</i> yang akan dipakai.</li> </ul>
5.	Kemampuan Sistem	Kemampuan sistem dalam hal ini dapat menampilkan <i>informasi</i> ODP berupa nama ODP, alamat, mitra, GPON dan status ODP secara cepat dan efisien

## Desain Aplikasi

Desain aplikasi bisa dilakukan setelah dilakukan analisis kebutuhan sistem langkah berikutnya dilakukan desain sistem menggunakan UML dan menggunakan bantuan Microsoft Visio diantaranya:

### 1. Use Case Diagram

*Use Case Diagram* bertujuan menggambarkan interaksi yang melibatkan sistem dengan pengguna sistem (*user*) atau bisa diartikan sebagai pemodelan untuk perlakuan sistem *informasi* yang dibuat (Mardian, et.al., 2021). *Use Case Diagram* merupakan proses penjabaran dari bentuk interaksi antara aktor dengan sistemnya, representasi antara interaksi sistem dengan aktor mempunyai tujuan agar mudah dipahami oleh *user* mengenai alur sistem dari aplikasi yang dibuat (Sutejo and Tanaamah. 2022). Tujuan *use case* diagram pada penelitian ini untuk dapat mengetahui pergerakan sistem *informasi* ODP pada PT Telkom Akses Samarinda.

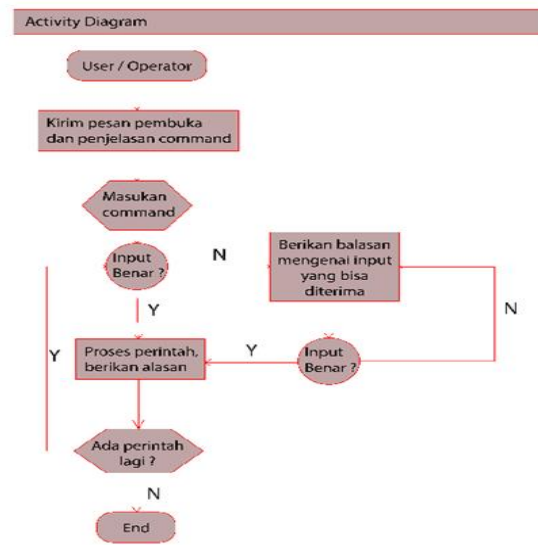


**Gambar 2.** Use Case Diagram

Pada gambar 2. Pengguna yaitu *user* atau operator dapat berinteraksi dengan sistem berupa *bot Telegram* dimana pengguna dapat mengelola data ODP yang maksud

## 2. Activity Diagram

*Activity Diagram* memuat informasi terkait bagaimana sistem bekerja terdapat beberapa aksi seperti menambah, mengedit, dan menghapus pesanan yang bisa dilakukan oleh *user*. *Activity diagram* merepresentasikan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi



**Gambar 3.** Activity Diagram Bot Telegram

Pada gambar 3. Menampilkan *activity diagram* pada halaman manajemen persediaan meliputi mulai, beberapa aktivitas untuk mengakses *bot* tersebut

## Implementasi Sistem

Implementasi dari sistem merupakan realisasi berdasarkan hasil rancangan yang sudah dibuat dengan UML berupa use case diagram dan *activity diagram* yang direalisasikan dengan bantuan *Apps script*. Tujuan implementasi sistem sendiri untuk memastikan sejauh mana *bot*

berjalan pada penelitian ini. Setiap pengguna yang akan memanfaatkan bot ini terlebih dahulu harus mendaftar admin dengan mengirimkan *id Telegram* mereka, apabila sudah, maka pengguna baru dapat mengakses bot.

### 1. Halaman Apps script yang digunakan pada bot

```
1 var token = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" : //  
2 var sheetID = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX" : //  
3 var welcomeMessage = "Selamat datang! Silakan masukkan nama ODP untuk mencari lokasinya."  
4  
5 function doPost(e) {  
6   var update = JSON.parse(e.postData.contents);  
7   if (update.message && update.message.text) {  
8     var chatId = update.message.chat.id;  
9     var messageText = update.message.text;  
10    var userId = update.message.from.id;  
11  
12    if (messageText == "/start") {  
13      sendText(chatId, welcomeMessage);  
14    } else {  
15      if (isUserAuthorized(userId)) {  
16        var odpDataSheet1 = findODPInSheet(messageText); // Menggunakan Sheet1  
17  
18        if (odpDataSheet1) {  
19          sendLocation(chatId, odpDataSheet1);  
20        } else {  
21          sendText(chatId, "ODP tidak ditemukan.");  
22        }  
23      } else {  
24        sendText(chatId, "Maaf, Anda tidak diizinkan mengakses bot ini.");  
25      }  
26    }  
27  }  
28 }  
29  
30 // Fungsi untuk mencari ODP dalam Sheet1 dengan cache  
31 function findODPInSheet(partialName) {  
32   var cache = CacheService.getScriptCache();  
33   var cachedResult = cache.get("odpCache_" + partialName);  
34  
35   if (cachedResult) {  
36     return JSON.parse(cachedResult);  
37   }  
38  
39   var sheet = SpreadsheetApp.openById(sheetID).getSheetByName("Sheet1");  
40   var data = sheet.getDataRange().getValues();  
41  
42   // Gunakan objek untuk menyimpan hasil pencarian  
43   var result = {  
44     odpName: null,  
45     latitude: null,  
46     longitude: null,  
47     address: null,  
48     partner: null,  
49     gpon: null,  
50     status: null  
51   };  
52  
53   for (var i = 1; i = data.length; i++) {  
54     var odpName = data[i][0].toLowerCase();  
55     if (odpName.indexOf(partialName.toLowerCase()) != -1) {  
56       // Setel hasil pencarian langsung ke objek result  
57       result = {  
58         odpName: data[i][0],  
59         latitude: data[i][1],  
60         longitude: data[i][2],  
61         address: data[i][3],  
62         partner: data[i][4],  
63         gpon: data[i][5],  
64         status: data[i][6]  
65       };  
66       // Hentikan pencarian setelah menemukan hasil pertama  
67       break;  
68     }  
69   }  
70 }  
71  
72 // Simpan hasil pencarian dalam cache selama 1 jam  
73 cache.put("odpCache_" + partialName, JSON.stringify(result), 3600);  
74  
75 return result.odpName ? result : null; // Kembalikan null jika tidak ditemukan  
76 }  
77  
78 // Fungsi untuk memeriksa apakah pengguna memiliki akses  
79 function isUserAuthorized(userId) {  
80   var sheet = SpreadsheetApp.openById(sheetID).getSheetByName("UserList");  
81   var data = sheet.getDataRange().getValues();  
82  
83   for (var i = 1; i = data.length; i++) {  
84     if (data[i][0] == userId) {  
85       return true;  
86     }  
87   }  
88  
89   return false;  
90 }  
91  
92 // Fungsi untuk mengirim pesan teks ke Telegram  
93 function sendText(chatId, text) {  
94   // Tambahkan spasi di awal setiap baris dan jarak satu kali enter  
95   text = text.replace(/[\r\n]/g, "\n ");  
96  
97   var payload = {  
98     method: "sendMessage",  
99     chat_id: chatId,  
100    text: text,  
101    parse_mode: "HTML",  
102  };  
103  sendTelegramRequest(payload);  
104 }  
105  
106 // Fungsi untuk mengirim lokasi ke Telegram dengan informasi tambahan  
107 function sendLocation(chatId, odpData) {  
108   var text = "Nama ODP : " + (odpData.odpName || "-") + "  
109     " + "Alamat : " + (odpData.address || "-") + "  
110     " + "Partner : " + (odpData.partner || "-") + "  
111     " + "Gpon : " + (odpData.gpon || "-") + "  
112     " + "Status : " + (odpData.status || "-")";  
113  
114   // Kirim pesan teks  
115   sendText(chatId, text);  
116  
117   // Kirim lokasi dengan tombol inline untuk melihat di Google Maps  
118   var payload = {  
119     method: "sendLocation",  
120     chat_id: chatId,  
121     latitude: odpData.latitude,  
122     longitude: odpData.longitude,  
123     reply_markup: {  
124       inline_keyboard: [  
125         [  
126           {  
127             text: "Lihat di Google Maps",  
128             url: "https://www.google.com/maps?d=" + odpData.latitude + "," + odpData.longitude,  
129             type: "url",  
130           }  
131         ]  
132       ]  
133     }  
134   };  
135   sendTelegramRequest(payload);  
136 }  
137  
138 // Fungsi untuk mengirim permintaan ke API Telegram  
139 function sendTelegramRequest(payload) {  
140   var token = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" : //  
141   var url = "https://api.telegram.org/bot" + token + "/" + payload.method;  
142   var options = {  
143     method: "POST",  
144     contentType: "application/json",  
145     payload: JSON.stringify(payload),  
146   };  
147   var response = UrlFetchApp.fetch(url, options);  
148 }
```

Gambar 4. Script yang digunakan pada bot



Pada gambar 4. Menunjukkan *apps script* yang terintegrasi dengan *bot*. Artinya semua pembuatan fungsi *bot* ditampilkan pada gambar

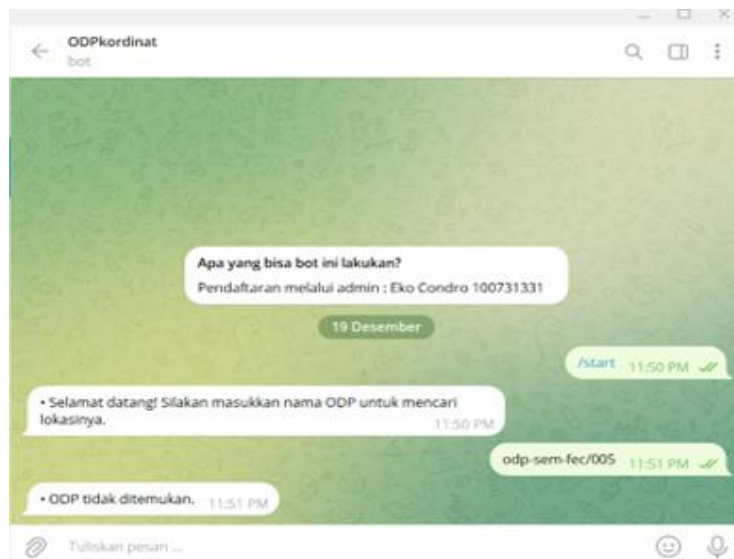
2. Halaman utama *bot* (apabila belum mendaftar)



**Gambar 5.** Halaman utama *bot* (apabila belum mendaftar)

Pada gambar 5. Merupakan tampilan halaman utama manajemen persediaan yang menampilkan kode, nama item, kemasan, satuan, stok tersedia, dan lokasi

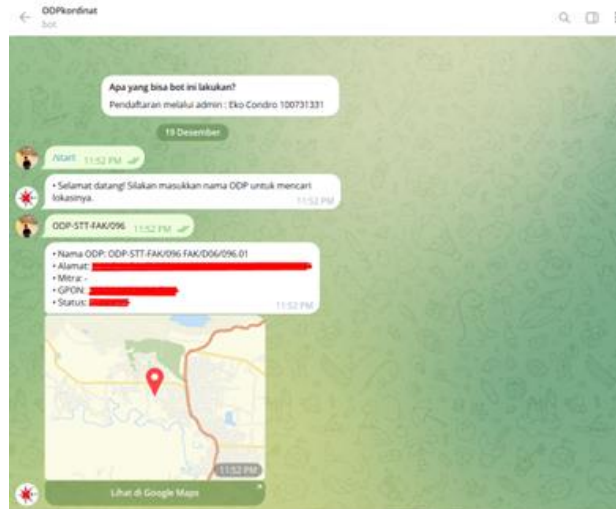
3. Halaman utama *bot* (apabila sudah mendaftar)



**Gambar 6.** Halaman utama *bot* (apabila sudah mendaftar)

Pada gambar 6. Menampilkan *form input* untuk menambahkan barang atau *item* pada manajemen persediaan dan sudah memiliki kode yang berbeda untuk setiap barang berupa kode, nama item, kemasan, stok tersedia, dan lokasi.

#### 4. Hasil tampilan ketikan mengakses *bot*



**Gambar 7.** Hasil tampilan ketikan mengakses *bot*

Pada gambar 7. Menampilkan informasi ODP berupa nama ODP, alamat, mitra, GPON dan status ODP.

#### **Pengujian (*Testing*) dan Verifikasi**

Proses terakhir adalah menguji Bot untuk menampilkan informasi kepada pengguna. Uji *bot* adalah aktivitas yang bertujuan untuk mengevaluasi atribut dan kemampuan bot dan memastikan bahwa sudah memenuhi hasil yang dicari. Penelitian ini dilakukan dengan dengan model *blackbox functional testing*. Metode ini menguji tanpa harus mengawasi kode sumber dari *website*. Metode ini bertujuan untuk memastikan fungsi dari bot berjalan sebagaimana mestinya. Ditunjukkan pada Tabel 2. Hasil pengujian *bot* dengan metode *blackbox functional testing* semua sistem berhasil dan tidak menunjukkan *error*.

**Tabel 2. Hasil Uji Black box Testing**

No	Pertanyaan	Status
1	Bot dapat menjalankan fungsi akses pengguna tertentu	Berhasil
2	Bot dapat menampilkan pesan sambutan untuk pengguna	Berhasil
3	Bot dapat menjalankan input ODP oleh pengguna	Berhasil
4	Bot dapat menampilkan informasi sesuai dengan input yang dilakukan	Berhasil

Tahap verifikasi hampir sama dengan pengujian (*testing*). Verifikasi dilakukan secara langsung oleh *admin* dan teknisi untuk memastikan kesesuaian informasi

#### **Pemeliharaan (*Maintenance*)**

Tahapan SDLC yang terakhir adalah proses *maintenance* atau pemeliharaan *bot*. Di tahap ini, tim akan melakukan pemeliharaan sistem dan rutin melakukan pembaruan agar kinerja *bot* tetap dapat optimal.

## KESIMPULAN DAN SARAN

Dengan memanfaatkan *Apps script* berbasis *JavaScript* yang terintegrasi langsung dengan data pada *Google sheets*, PT. Telkom Akses Samarinda berhasil mengembangkan *bot* ODP Kordinat. Keberhasilan ini membawa dampak positif bagi kinerja administratif dan teknis, memudahkan admin dalam memperbarui data dan membantu teknis dalam pencarian informasi ODP yang dibutuhkan. Penerapan teknologi ini telah terbukti menjadi pilihan yang tepat untuk mempercepat proses kerja tim teknis dan tim kantor dalam penyampaian informasi ODP.

Pentingnya kemudahan akses dan pembaruan data melalui *bot* ini juga memberikan dampak besar bagi perusahaan secara keseluruhan. Saran untuk penelitian selanjutnya adalah mengkaji kemungkinan implementasi *bot* dengan sistem berbayar. Hal ini dapat memberikan kontribusi positif dalam meningkatkan fitur, melakukan upgrade sistem, dan memperkenalkan fungsionalitas baru pada *bot*. Peningkatan ini bertujuan untuk meningkatkan kinerja *bot* dengan sistem yang lebih mutakhir.

Selain itu, pengembangan *bot* tidak hanya sebatas pada divisi teknis, melainkan dapat diperluas untuk digunakan di divisi-divisi lain di PT Telkom Akses Samarinda sesuai dengan kebutuhan masing-masing. Dengan demikian, teknologi ini dapat menjadi solusi yang lebih holistik dan dapat diadopsi secara luas di berbagai bidang operasional perusahaan.

## DAFTAR REFERENSI

- A. J. Sutejo and A. R. Tanaamah, "Perancangan dan Implementasi Sistem Informasi Pendataan Barang dengan Aplikasi WDCSI 'Warehouse Data Collection with System Information,'" *Aiti*, vol. 19, no. 1, pp. 103–119, 2022, [Online]. Tersedia: <https://ejournal.uksw.edu/aiti/article/view/6185>
- A. Mardian, B. Thomas, H. Rachmawaty, and Y. Verdi, "Perancangan Aplikasi Pemantauan Kinerja Karyawan Berbasis Android Di PT. Salestrade Corp. Indonesia," *J. Manaj. Inform. Jayakarta*, vol. 1, pp. 169–185, 2021, [Online]. Available: <https://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta/article/view/481>.
- Aditya P. S. 2023. Rancang Bangun Sistem Informasi Approval Dismantling NTE Telkom Akses dengan Metode RAD (Rapid Application Development). Skripsi. Fakultas Ilmu Komputer. Universitas Pembangunan Panca Budi: Medan.
- Alfariz, S. 2021. Keahlian Teknik Jaringan Akses Telekomunikasi. Kelas XII. Batam: Laporan Kegiatan Praktik Kerja Lapangan pada PT Telkom Akses Witel Riau Kepulauan (19 November 2021)

- B. N. Prastowo, N. A. S. Putro and O. A. Dhewa, "PLO User Interface Based on *Telegram Bot*," Indonesian Journal of Computing and Cybernetics Systems, vol. 13, no. 1, pp. 21-30, 2019.
- Bator, R. J., Bryan, A. D., & Schultz, P. W. (2011). Who Gives a Hoot?: Intercept Surveys of Litterers and Disposers. *Environment and Behavior*, 43(3), 295–315. <https://doi.org/10.1177/0013916509356884>.
- Fatmala, N.A., and Azizah, N. "Sistem Aplikasi Dan Pengerjaan Order Pada PT Telkom Akses Malang Divisi Drafter Dan SDI (Survey Drawing Inventory)" *Jurnal ilmiah Sistem Informasi dan Ilmu Komputer*. Volume 3, No.1, pp113-121. Link: <https://journal.sinov.id/index.php/juitik/article/view/504>
- Hamdan, A. A, dkk. 2018. *Teknik Informatika*. Malang: Laporan Praktik Kerja Lapangan (PKL) Perusahaan/Industri PT Telkom Akses Yogyakarta Implementasi Jaringan Outside Plant pada Arsitektur Fiber Optic To The Home
- I. Jayusman and O. A. K. Shavab, "Aktivitas Belajar Mahasiswa Dengan Menggunakan Media Pembelajaran Learning Management System (Lms) Berbasis Edmodo Dalam Pembelajaran Sejarah," *J. Artefak*, vol. 7, no. 1, p. 13, 2020, doi: 10.25157/ja.v7i1.3180, [Online]. Tersedia: <https://jurnal.unigal.ac.id/artefak/article/view/3180>.
- J. Gregorio, A. Gardel and B. Alarcos, "Forensic Analysis of *Telegram* Messenger for Windows Phone," *Digital Investigation*, pp. 88-106, 2017.
- Mahmud, M., Purnawansyah., Hasnawi, M." Implementasi *Bot Telegram* Untuk Monitoring Jaringan Dengan Pendekatan Security Policy Development Life Cycle Pada Kementerian Kelautan dan Perikanan Untia" *Buletin Sistem Informasi dan Teknologi Islam*, Vol.3 No.2, pp127-133, Link: <https://jurnal.fikom.umi.ac.id/index.php/BUSITI/article/view/1162>
- Mulyanto, A.D., "Pemanfaatan *Bot Telegram* Untuk Media Informasi Penelitian" *Jurnal Ilmu Komputer dan Teknologi Informasi*. Volume 12, No. 1 (2020), pp 49-54. Link: <https://ejournal.uin-malang.ac.id/index.php/saintek/article/view/8847/7727>.
- Mulyanto, A.D., "Pemanfaatan *Bot Telegram* Untuk Media Informasi Penelitian" *Jurnal Ilmu Komputer dan Teknologi Informasi*. Volume 12, No. 1 (2020), pp 49-54. Link: <https://ejournal.uin-malang.ac.id/index.php/saintek/article/view/8847/7727>
- Ningrat, W., (2016). Perancangan Jaringan Distribusi Fiber To The Home (FTTH) di Komplek Batununggal Indah Bandung. In *MEETAS: Modern Electrical Engineering Technology and Its Application Seminar Vol. 2 No. 1*. Serat Optik [2,3] (hlm 1-10).
- Prabowo, Y., Aulia, S. Maulana, R. "Pembuatan *Bot Telegram* Untuk Teknisi Assurance Dengan Metode Push Notification Di Pt Telkom Akses Rajawali Kota Bandung" e-Proceeding of Applied Science. Vol.9 No.1, pp113-118. Link: <https://openlibrary.telkomuniversity.ac.id/home/catalog/id/182344/slug/pembuatan-bot-Telegram-untuk-teknisi-assurance-dengan-metode-push-notification-di-pt-telkom-akses-rajawali-kota-bandung.html>

- Romdani, F. 2022. “Analisis dan Perancangan Sistem Informasi Geografis Pemetaan Lokasi Optical Distribution Cabinet (ODC) dengan Metode Rapid Application Development (RAD) di PT Telkom Akses Bintaro” dalam Jurnal Esit (E-Bisnis, Sistem Informasi, Teknologi Informasi) Vol. 17, No 2 (hlm. 1-8). Tangerang Selatan: Open Journal Universitas Pamulang Indonesia
- S. R. Zulfitra and A. Ayuningtyas, “Aplikasi Manajemen Risiko SPBE berbasis Website pada Dinas Komunikasi dan Informatika Kabupaten Gresik,” J. Teknol. dan Inf., vol. 13, no. 2, pp. 138–151, 2023, [Online]. Tersedia: <https://ojs.unikom.ac.id/index.php/jati/article/view/9484>.
- Teniwut, M. 2022. “Teknik Pengumpulan Data dan Metode Penelitian”, <https://mediaindonesia.com/humaniora/539107/teknik-pengumpulan-data-dan-metodepenelitian>, Diakses pada 25 Desember 2023
- Website PT Telekomunikasi Indonesia Tbk. “PraBac-2015 Dashboard Telkom.” <https://dashboard.telkom.co.id/>, diakses pada 23 Desember 2023
- Y. P. Atmojo, "Bot Alert Snort dengan Telegram Bot API pada Intrusion Detection System: Studi Kasus IDS pada Server Web," in Seminar Nasional Sistem Informasi dan Teknologi Informasi 2018, Pontianak, 2018