

# BAB I

## PENDAHULUAN

### 1.1 LATAR BELAKANG

*Mobile* merupakan peningkatan dari sistem perangkat lunak terpadu yang umumnya ditemukan pada *Personal Computer Desktop*. Pada awal kemunculannya, aplikasi menyediakan fungsionalitas yang terbatas dan terisolasi seperti permainan, kalkulator atau *mobile Web browsing* sehingga aplikasi sangat dihindari karena multitasking yang ‘memakan’ sumber daya *hardware* perangkat *mobile* yang awalnya cukup terbatas. Namun, pada jaman sekarang ini dengan dukungan sumber daya *hardware* yang lebih tinggi, aplikasi telah menjadi hal yang tak terpisahkan dengan perangkat *mobile* sebab mereka mampu melakukan apapun dengan mudah.

PT. Telkom Akses merupakan salah satu anak perusahaan dari PT. Telkom Indonesia yang bergerak di bidang *manage service* dan konstruksi pembangunan infrastruktur jaringan komunikasi. PT. Telkom Akses didirikan pada tanggal 12 Desember 2012. Saham pada PT. Telkom Akses sepenuhnya dimiliki oleh PT. Telkom Indonesia, Tbk. PT Telkom Akses didirikan sebagai bagian dari komitmen Telkom untuk terus melakukan pengembangan jaringan *broadband* untuk menghadirkan akses informasi dan komunikasi tanpa batas bagi seluruh masyarakat Indonesia. Telkom berupaya menghadirkan koneksi internet berkualitas sekaligus terjangkau untuk meningkatkan kualitas sumber daya

manusia (SDM) sehingga mampu bersaing di tingkat dunia. Saat ini Telkom tengah membangun jaringan *backbone* berbasis Serat Optik maupun *Internet Protocol* (IP) dengan menggelar 30 *node terra router* dan sekitar 75.000 Km kabel Serat Optik.

Pada perusahaan ini terdapat banyak aktifitas kerja salah satunya adalah tentang pengolahan data. Untuk pengolahan data sendiri dilakukan antar berbagai divisi atau unit kerja. Secara umum unit kerja yang terkait dibagi menjadi dua yaitu unit kerja yang berada di lapangan (*onsite*) dan unit kerja yang berada di *office* (*ondesk*). Untuk saat ini koordinasi antar divisi dilakukan dengan menggunakan *platform* telegram. Alur aktivitas kerja secara umum dapat di gambarkan sebagai berikut divisi yang di lapangan meminta data ke divisi yang berada di *office* setelah itu divisi yang di *office* mencari data yang diminta tersebut di suatu *database* yang disimpan pada *Google Spreadsheets* dan mengirimkannya kembali ke divisi lapangan, data-data yang dibutuhkan berupa data ketersediaan alat produksi di suatu koordinat tertentu dan profil dari suatu alat produksi. Alur kerja tersebut secara umum cukup baik tetapi dengan penerapan teknologi yang ada maka alur kerja tersebut dapat di maksimalkan. Adapun penerapan teknologi untuk mempermudah aktivitas kerja adalah dengan menggunakan telegram bot. Telegram bot adalah sebuah bot atau robot yang diprogram dengan berbagai perintah untuk menjalankan serangkaian instruksi yang diberikan oleh pengguna. Bot ini hanyalah sebuah akun Telegram yang dioperasikan oleh perangkat lunak yang memiliki fitur *Artificial Intelligence*. Bot Telegram dapat

melakukan apa saja sesuai perintah (yang sudah tersedia). Bot telegram bisa digunakan untuk melakukan pencarian, sebagai penghubung, pengingat, pengajar, pengintegrasian, dan lainnya. Dengan mengkolaborasi antara *database* yang berupa *Google Spreadsheet* dan bot telegram dengan algoritma tertentu maka dapat dijadikan sebagai *helpdesk online* yang dapat diakses oleh semua pengguna telegram. Selain itu pada penelitian ini akan dilakukan perhitungan tentang kebutuhan untuk membangun Bot Telegram dan dilakukan juga perhitungan tentang efektifitas dan efisiensi antara sistem lama dengan sistem yang baru dengan menggunakan beberapa indikator.

## **1.2 PERUMUSAN MASALAH**

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat dirumuskan permasalahan yaitu:

1. Bagaimanakah menganalisis dan membangun *Artificial Intelligence helpdesk online* dengan menggunakan *Bot Telegram* dan *Google Spreadsheet*?

## **1.3 BATASAN MASALAH**

Adapun batasan masalah dalam penelitian ini adalah:

1. Data yang digunakan dalam penelitian ini adalah data alat produksi atau disebut dengan ODP yang tersimpan dalam *Google Spreadsheet*.
2. Data yang ditampilkan berupa data yang terdapat dalam *Google Spreadsheet*.
3. Berfokus pada aplikasi telegram yang berupa bot telegram.

4. Menampilkan data jarak ODP terdekat, profil ODP, ketersediaan ODP.
5. Menghitung tingkat keefektifan sistem yang lama dengan Bot Telegram.

#### **1.4 TUJUAN PENELITIAN**

Adapun tujuan dari penelitian ini adalah sebagai berikut

1. Menganalisis dan membangun *Artificial Intelligence helpdesk online* dengan menggunakan *Bot Telegram* dan *Google Spreadsheet*

#### **1.5 MANFAAT PENELITIAN**

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Penelitian ini diharapkan mempermudah unit kerja yang berada di lapangan untuk memperoleh data secara *mobile*.
2. Penelitian ini diharapkan memaksimalkan akses terhadap data yang telah tersedia.
3. Penelitian ini diharapkan dapat mengoptimalkan proses kerja yang sudah ada

## **BAB II**

### **LANDASAN TEORI**

Pada penelitian ini terdapat berbagai landasan teori, adapun landasan teori yang digunakan pada penelitian ini adalah sebagai berikut.

#### **2.1. SERAT OPTIK**

(Mauludin, 2017) Serat optik adalah suatu saluran transmisi sejenis kabel yang terbuat dari bahan kaca atau plastik yang sangat halus serta lebih kecil dari sehelai rambut. Serat optic dapat digunakan untuk mentransmisikan sinyal cahaya dari suatu tempat ke tempat lain. Sumber cahaya yang digunakan adalah laser atau LED. Kabel ini berdiameter kurang lebih 120 mikrometer. Cahaya yang ada di dalam serat optik tidak keluar karena indeks bias dari kaca lebih besar daripada indeks bias dari udara, karena laser mempunyai spektrum yang sangat sempit. Kecepatan transmisi serat optik sangat tinggi sehingga sangat bagus digunakan sebagai saluran komunikasi. Perkembangan teknologi serat optik saat ini, telah dapat menghasilkan pelemahan (*attenuation*) kurang dari 20 *decibels* (dB)/km. Dengan lebar jalur (*bandwidth*) yang besar sehingga kemampuan dalam mentransmisikan data menjadi lebih banyak dan cepat dibandingkan dengan penggunaan kabel konvensional. Dengan demikian serat optik sangat cocok digunakan terutama dalam aplikasi sistem telekomunikasi. Pada prinsipnya serat optik memantulkan dan membiaskan sejumlah cahaya yang merambat didalamnya. Efisiensi dari

serat optik ditentukan oleh kemurnian dari bahan penyusun gelas/kaca. Semakin murni bahan gelas, semakin sedikit cahaya yang diserap oleh serat optik.

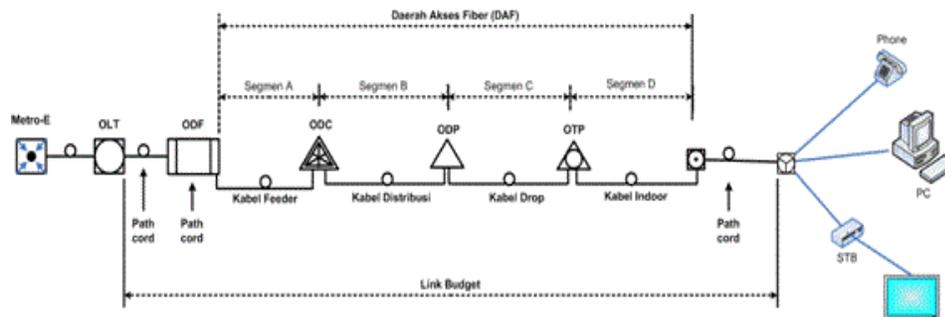
## **2.2. FTTH (FIBER TO THE HOME)**

(Wida Ningrat, 2016) *Fiber to the Home* disingkat FTTH merupakan suatu standar penghantaran informasi dari pusat penyedia (*provider*) ke kawasan pengguna dengan menggunakan penghantar serat optik sebagai medium. Teknologi ini berkembang tidak lepas dari kemajuan perkembangan serat optik yang dapat menggantikan penggunaan kabel konvensional. Dan juga didorong oleh suatu keinginan untuk mendapatkan layanan yang dikenal dengan istilah *Triple Play Services* yaitu layanan akses internet yang cepat, suara (jaringan telepon, PSTN) dan video (TV Kabel) dalam satu infrastruktur pada unit pelanggan. Penghantaran dengan menggunakan teknologi FTTH ini dapat menghemat biaya dan mampu mengurangi biaya operasi dan memberikan pelayanan yang lebih baik kepada pelanggan. Ciri-ciri inherent serat optik membenarkan penghantaran isyarat telekomunikasi dengan lebar jalur yang lebih besar dibandingkan dengan penggunaan kabel konvensional.

Dari gambar mengilustrasikan arsitektur umum dari suatu jaringan FTTH. Biasanya jarak antara pusat layanan dengan pelanggan dapat berkisar maksimum 20 km. Dimana pusat penghantaran penyelenggara layanan (*service provider*) yang berada di kantor utama disebut juga dengan *central office* (CO), disini terdapat peralatan yang disebut dengan OLT. Kemudian dari OLT ini dihubungkan kepada ONU yang ditempatkan di rumah-rumah pelanggan

(customer's) melalui jaringan distribusi serat optik (*Optical Distribution Network*, ODN). Isyarat optik dengan panjang gelombang (*wavelength*) 1490 nm dari hilir (*downstream*) dan isyarat optik dengan panjang gelombang 1310 nm dari hulu (*upstream*) digunakan untuk mengirim data dan suara. Sedangkan layanan video dikonversi dahulu ke format optik dengan panjang gelombang 1550 nm oleh optik pemancar video (*optical video transmitter*). Isyarat optik 1550 nm dan 1490 nm ini digabungkan oleh pengabung (*coupler*) dan ditransmisikan ke pelanggan secara bersama. Singkatnya, tiga panjang gelombang ini membawa informasi yang berbeda secara simultan dan dalam berbagai arah pada satu kabel serat optik yang sama. Secara umum topologi jaringan FTTH dapat dibagi menjadi 4 segmen catuan kabel yaitu sebagai berikut:

1. Segmen A: Catuan dengan istilah kabel Feeder.
2. Segmen B: Catuan dengan istilah kabel Distribusi.
3. Segmen C: Catuan dengan istilah kabel Penanggal / Drop.
4. Segmen D: Catuan dengan istilah kabel Rumah/Gedung.



Gambar 2.1 Topologi Jaringan FTTH

ODC (*Optical Distribusi Cabinet*) memiliki fungsi sebagai tempat instalasi sambungan jaringan optik (Gambar 2). Adapun kapasitas dari ODC adalah 96, 144, dan 288 port. Komponen-komponen yang ada dalam ODC sebagai berikut:

1. Kabel Tray berfungsi untuk menempatkan hasil sambungan kabel serat optik.
2. Konektor, ujung serat optik yang akan disambungkan pada konektor adaptor parking-lot, suatu tempat terminasi sementara konektor yang belum disambungkan.
3. Patch-cord merupakan kabel serat optik dengan panjang tertentu yang sudah dilengkapi dengan konektor di ujung kabel dan digunakan untuk menghubungkan antar perangkat.
4. Pig-tail adalah seutas kabel yang hanya memiliki satu buah konektor diujungnya. Pigtail akan disambungkan dengan kabel fiber yang belum memiliki konektor.
5. Slack storage digunakan untuk mengamankan dan melindungi kabel.
6. Splice Tray untuk melindungi sambungan fiber yang menggunakan teknik splicing.
7. Splice merupakan sambungan permanen antara dua serat optik.
8. Splitter perangkat yang digunakan untuk membagi sebuah sinyal optik.



Gambar 2.2 Optical Distribution Cabinet (ODC)

ODP (Optical Distribution Point) sebagai tempat terminasi atau penyambungan kabel dan memiliki bahan yang bersifat tahan korosi, dan tahan cuaca adapun bentuk dari ODP itu sendiri dapat dilihat pada Gambar 2. ODP memiliki fungsi sebagai tempat instalasi sambungan terutama untuk menghubungkan kabel distribusi dan kabel drop. Beberapa komponen yang terdapat pada ODP antara

lain *optical pigtail*, Konektor adaptor, *Splitter*. Ada beberapa jenis ODP yang sering digunakan, yaitu ODP Pole (Gambar 2.3) dan ODP Closure. (Gambar 2.4).



Gambar 2.3 Optical Distribution Point (ODP) Pole



Gambar 2.4 Optical Distribution Point (ODP) Closure

### 2.3. SISTEM INFORMASI GEOGRAFIS (SIG)

(Purmadipta, Anra and Irwansyah, 2016) Sistem Informasi Geografis (bahasa Inggris: *Geographic Information System disingkat GIS*) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Atau dalam arti yang lebih sempit, adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi

menurut lokasinya, dalam sebuah *database*. Para praktisi juga memasukkan orang yang membangun dan mengoperasikannya dan data sebagai bagian dari sistem ini. *Teknologi Sistem Informasi Geografis* dapat digunakan untuk investigasi ilmiah, pengelolaan sumber daya, perencanaan pembangunan, kartografi dan perencanaan rute. Misalnya, SIG bisa membantu perencana untuk secara cepat menghitung waktu tanggap darurat saat terjadi bencana alam, atau SIG dapat digunakan untuk mencari lahan basah (*wetlands*) yang membutuhkan perlindungan dari polusi.

#### **2.4. KARTOGRAFI**

(Cahyono and Zulkarnain, 2018) Kartografi (atau pembuatan peta) adalah ilmu dan praktik dalam membuat *globe* atau peta. Dengan kata lain kartografi selalu berhubungan serta membahas secara khusus tentang pembuatan peta serta interpretasinya. Peta secara tradisional sudah dibuat menggunakan pena dan kertas, tetapi munculnya dan penyebaran komputer sudah merevolusionerkan kartografi. Peta secara tradisional sudah dibuat menggunakan pena dan kertas, tetapi munculnya dan penyebaran komputer sudah merevolusionerkan kartografi. Banyak peta komersial yang bermutu sekarang dibuat dengan perangkat lunak pembuatan peta yang merupakan salah satu di antara tiga macam utama; CAD (desain berbantuan komputer), GIS (Sistem Informasi Geografis), dan perangkat lunak ilustrasi peta yang khusus.

Peta juga dibedakan menjadi 2 jenis yaitu:

- Peta Garis adalah peta topografi dan peta tematik

- Peta Foto adalah peta photo dan peta orthophoto

Adapun beberapa fungsi peta diantaranya:

- Menunjukkan lokasi dan posisi relatif
- Memperlihatkan ukuran (Luas, Jarak dan Isi) -Memperlihatkan bentuk (Benua Gunung, benua, negara dll)
- Mengumpulkan dan menyeleksi data dari suatu daerah dan menyajikannya di atas peta.

Skala yang digunakan juga dapat berupa skala numerik maupun skala batang (grafis). Kartografi berkembang dari kumpulan teknik menggambar menjadi sebuah ilmu. Seorang kartografer harus memahami psikologi kognitif dan ergonomic untuk membuat symbol yang cocok untuk mewakili informasi tentang bumi sehingga bisa dimengerti orang lain secara efektif. Kartografer juga perlu memahami psikologi perilaku untuk mempengaruhi pembaca agar memahami informasi yang dibuatnya. Mereka juga harus mempelajari ilmu geodesi dan matematika yang tidak sederhana untuk memahami bagaimana bentuk bumi berpengaruh terhadap penyimpangan atau distorsi dari proses proyeksi ke bidang datar. Kartografi adalah seni, ilmu pengetahuan dan teknologi tentang pembuatan peta-peta sekaligus mencakup studinya sebagai dokumen-dokumen ilmiah dan hasil karya seni (ICA,1973). Kartografi merupakan sebuah teknik pembuatan peta yang secara mendasar berhubungan dengan memperkecil keruangan pada suatu daerah yang luas di permukaan bumi atau benda luar angkasa yang di sajikan dalam bentuk yang mudah di fahami sehingga dapat di gunakan untuk kepentingan komunikasi bagi khalayak ramai.

## 2.5. ARTIFICIAL INTELLIGENCE

*Artificial Intelligence* atau AI merupakan salah satu teknologi yang sedang populer saat ini. Berbagai bidang industri sudah memanfaatkan teknologi tersebut, mulai dari keuangan, kesehatan, penjualan, pertanian, keamanan dan lain lain. Selain itu, *Artificial Intelligence* juga sudah banyak diterapkan di berbagai kehidupan sehari-hari. *Artificial Intelligence* banyak membantu dalam berkomunikasi, menemukan lokasi, menemukan pola, dan membantu memprediksi sesuatu. *Artificial Intelligence* atau kecerdasan buatan adalah sistem komputer yang mampu melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Teknologi ini dapat membuat keputusan dengan cara menggunakan data yang tersedia di dalam sistem dan dianalisis menggunakan algoritma tertentu. Proses yang terjadi dalam *Artificial Intelligence* mencakup *learning*, *reasoning*, dan *self-correction*. Proses ini sangat mirip dengan manusia yang melakukan analisis sebelum memberikan suatu keputusan. Menurut John Mc Carthy, 1956, *Artificial Intelligence* adalah untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan dan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik. Keinginan untuk mencapai situasi yang memberikan kemampuan kepada perusahaan agar tetap tumbuh dan berkembang serta mampu menghasilkan keuntungan yang wajar dimotivasi oleh kekhawatiran yang sering menghantui pikiran para pebisnis

sehubungan dengan perubahan dinamis pada lingkungan industri yang memasuki revolusi Industri 4.0. Pada era ini permintaan terhadap produk yang sesuai dengan keinginan konsumen dan dapat di peroleh dengan waktu yang relatif singkat menjadi tema utama dalam pengembangan produk. Faktor biaya per unit minimum dan mutu produk yang tinggi tidak lagi memiliki kekuatan untuk membangun daya saing. Daya saing ditentukan oleh kinerja dalam pengiriman, kemampuan memenuhi keinginan konsumen dan kualitas keterlibatan dalam penanganan isu-isu lingkungan (Sukaria, 2008).

## **2.6. BOT TELEGRAM**

Seperti namanya, bot merupakan istilah yang diambil dari kata robot yang digunakan untuk mengoperasikan sebuah akun tanpa pengguna atau non-manusia. Salah satu yang menggunakan cara ini yaitu aplikasi Telegram. Cara ini terdiri dari garis kode dan mampu melakukan tindakan yang cukup kompleks dengan otomatis. Selain itu, ketika menggunakan bot Telegram juga menerima perintah pengguna dalam bentuk permintaan HTTP. Dalam pengaplikasiannya, bot Telegram sama saja seperti akun pribadi Telegram pada umumnya. Namun, pada akun bot tidak mempunyai nomor telepon dan digunakan oleh komputer. Selain itu, bot hanya bisa merespon perintah sesuai dengan parameter yang telah ditentukan oleh admin bot. Kegunaan bot telegram diantaranya:

1. Bot dapat digunakan sebagai koran pintar (*smart newspaper*) yang akan memberikan berita kepada pelanggan bot tersebut.

2. Bot juga dapat digunakan sebagai jembatan layanan lain seperti Gmail, Gambar, GIF, IMDB, Wiki, Musik, Youtube, GitHub.
3. Bot dapat juga digunakan untuk menerima pembayaran dari pengguna telegram.

## **2.7. GOOGLE SPREADSHEET**

*Google Spreadsheets* adalah sebuah aplikasi *spreadsheet online* yang memungkinkan Anda membuat dan Format *spreadsheet* dan sekaligus bekerja sama dengan orang lain. *Google Drive Spreadsheet*, aplikasi layanan pengolah dokumen, terutama untuk dokumen keuangan atau tabel yang terlihat sederhana, terus dikembangkan oleh *Google*. Hasilnya, aplikasi tersebut kini dapat melakukan pengolahan dokumen yang bersifat analisis. *Google* pun mengubah nama aplikasi tersebut menjadi *Google Sheets* dan menyematkan beragam fitur menarik di dalamnya. *Google* menjanjikan, *Google Sheets* akan dapat bekerja lebih cepat, mampu menangani beragam dokumen yang penuh dengan angka-angka, dan dapat digunakan secara *offline*.

## **BAB III**

### **METODOLOGI PENELITIAN**

Penelitian ini akan dilaksanakan berdasarkan tahapan sebagai berikut ini

#### **3.1. STUDI LITERATUR**

Mempelajari literatur dan tinjauan pustaka seperti jurnal, paper, artikel, dan buku atau ebook yang relevan dengan penelitian ini yaitu meliputi topik *Google Spreadsheet, Javascript* dan Bot Telegram

#### **3.2. COLLECTING DATA**

Tahapan yang dilakukan untuk mencari sumber data yang akan di analisis, adapun metode pengumpulan data yang di lakukan adalah dengan melakukan observasi. Observasi memungkinkan untuk mengambil data dengan cara mengamati secara langsung. Adapun data-data yang berhasil didapat adalah sebagai berikut kumpulan data disimpan di dalam *Google Spreadsheet* Data tersebut berisi informasi tentang alat produksi atau alpro atau odp seluruh Lampung dengan *field* input artinya *field* di input secara manual ditandai dengan warna hijau muda. Berikut ini penjelasan dari *field* input:

1. ODP/ODC/STO adalah *field* yang berisi nama ODP atau ODC atau STO yang berada di wilayah Lampung.
2. TIKOR adalah *field* yang berisi titik koordinat dengan format desimal x,y tempat ODP atau ODC atau STO berada.

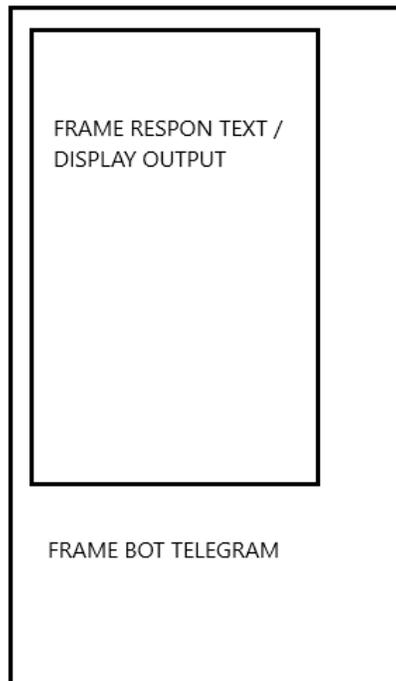
3. DIST adalah *field* yang berisi tempat keberadaan distribusi suatu ODP.
4. TANGGAL GOLIVE adalah *field* yang berisi tanggal ODP tersebut masuk ke sistem.
5. TYPE adalah *field* yang berisi tipe project pada suatu ODP atau ODC terdiri dari PT2, PT2-Project, PT3, PT4
6. NAMA LOP adalah *field* yang berisi nama project pada suatu ODP atau ODC yang dibangun
7. CORE DIST adalah *field* yang berisi core distribusi suatu ODP
8. CORE FEE adalah *field* yang berisi core feeder suatu ODP
9. IP adalah *field* yang berisi IP OLT dari suatu ODP
10. MODUL adalah *field* yang berisi Modul OLT dari suatu ODP
11. PORT adalah *field* yang berisi Port OLT dari suatu ODP
12. MITRA adalah *field* yang berisi nama mitra pembangun ODP atau ODP
13. ID SW adalah *field* yang berisi ID dari aplikasi Small World yang akan di input ke sistem.
14. JUMLAH PANEL adalah *field* yang berisi jumlah panel yang terdapat di ODP

### **3.3. PERANCANGAN SISTEM**

Tahapan ini merupakan perancangan aplikasi adapun aplikasi ini dirancang dengan *platform BOT Telegram* dan *Google Spreadsheet*. Platform ini dipilih karena aplikasi ini ditujukan untuk dapat di akses secara mobile. Berikut ini adalah tahapan perancangan sistem

## 1. Perancangan Output

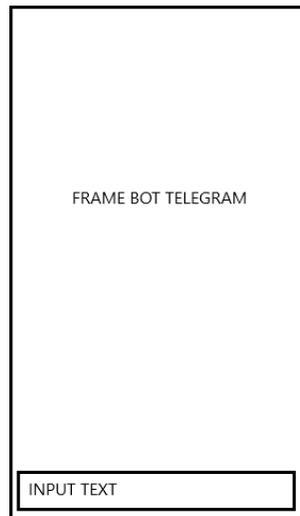
Pada penelitian ini output data dirancang berupa informasi balasan terhadap respon yang diminta oleh *user*. Data balasan tersebut berisi data profil suatu ODC atau ODP atau STO yang berada di Lampung, respon juga dapat berupa daftar ODC atau ODP atau STO yang terdekat



Gambar 2.5 Perancangan Output

## 2. Perancangan Input

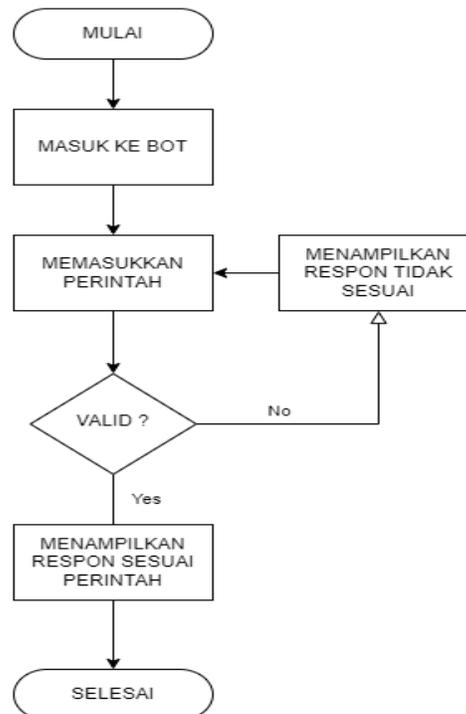
Pada penelitian ini input dirancang berupa perintah atau data chatting yang di masukkan *user* ke dalam Akun BOT Telegram.



Gambar 2.6 Perancangan Input

### 3. Perancangan Proses Sistem

Berikut ini perancangan proses Sistem Bot Telegram yang di gambarkan dalam Gambar 2.7



Gambar 2.7 Flowchart Bot Telegram

- a. *User* masuk ke Bot Telegram
- b. *User* memasukkkan perintah ke Bot Telegram dengan standard perintah tertentu.
- c. Bot Telegram memvalidasi perintah tersebut dengan mengecek data ke *database* yang ada di *Google Spreadsheet*
- d. Jika perintah yang di masukkan tidak sesuai maka Bot telegram akan menampilkan respon tidak sesuai atau perintah salah.
- e. Jika perintah yang di masukkan sesuai maka Bot telegram akan menampilkan respon berupa data yang diminta *User*.

#### 4. Perancangan *Database*

*Database* disimpan di dalam *Google Spreadsheet* dan beberapa data atau *field* dimanipulasi dengan formula untuk mendapatkan informasi yang lebih berguna. Berikut ini keterangan dari setiap *field* yang diberikan formulasi.

- a. CEK DUPLIKASI adalah *field* yang berfungsi untuk mengecek duplikasi data pada setiap data yang di input jika ada data yang duplikasi *field* ini akan menghitung jumlah data yang sama
- b. STO adalah *field* yang berisi nama STO yang berada di Lampung dengan mengambil data
- c. MEREK OLT adalah *field* yang berisi merek OLT yang berada di Lampung berdasarkan sheet Info Olt yang diambil dari *field* IP
- d. POWER OLT adalah *field* yang berisi power OLT yang berada di Lampung berdasarkan sheet Info Olt yang diambil dari *field* IP

- e. FORMULASI adalah *field* yang berisi nama ODC/Distribusi berdasarkan *field* ODP/ODC/STO dan *field* DIST
- f. LATITUDE adalah *field* yang berisi titik kordinat garis bujur dengan format decimal yang diambil dari *field* TIKOR
- g. LONGITUDE adalah *field* yang berisi titik kordinat garis lintang dengan format decimal yang diambil dari *field* TIKOR

#### **3.4. IMPLEMENTASI SISTEM**

Tahapan ini merupakan tahapan membangun sistem dengan analisis dan perancangan yang telah dilakukan. Tahap impementasi sistem merupakan tahap menerapkan sistem supaya sitem tersebut dapat dioperasikan. Tahap ini termasuk juga kegiatan menulis kode program dan menulis algoritma yang ditanam pada sistem.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1. ANALISA KEBUTUHAN**

Kebutuhan yang diperlukan untuk penelitian ini adalah sebagai berikut:

##### 1. Kebutuhan Hardware

Satu unit Notebook dengan spesifikasi sebagai berikut:

Processor: Intel Core i7-10750H

RAM: 8 GB DDR4

Storage: 512 GB SSD

VGA: Nvidia GeForce GTX 1650 Ti Max-Q 4 GB

Display: 15.6-inch Full HD IPS 144 Hz

##### 2. Kebutuhan Software

Software yang dibutuhkan untuk penelitian ini adalah

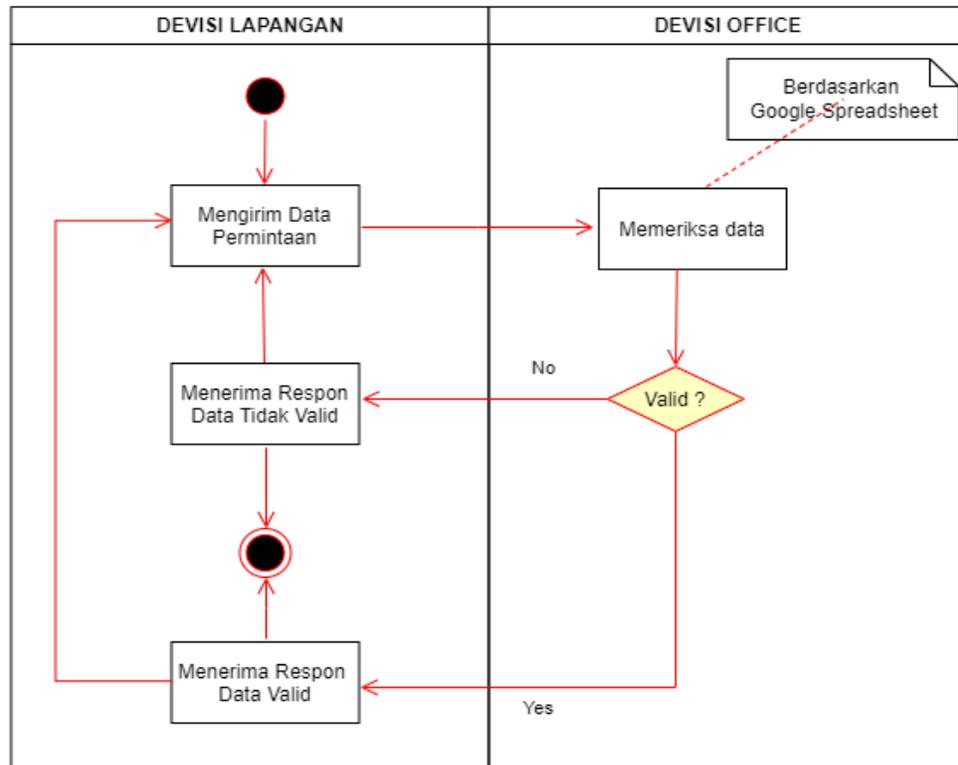
- a. Sistem Operasi Windows 10
- b. *Google Chrome*
- c. *Google Spreadsheet*
- d. Telegram

#### **4.2. ALUR KERJA SISTEM**

Alur kerja sistem adalah tahapan yang menggambarkan suatu sistem bekerja, pada penelitian ini alur kerja sistem dibagi menjadi dua yaitu sebagai berikut:

## 1. Alur Sistem Lama

Alur sistem ini menggambarkan alur sistem yang digunakan saat sistem baru belum di gunakan. Adapun entitas yang terlibat dapat digambarkan pada activity diagram di bawah ini



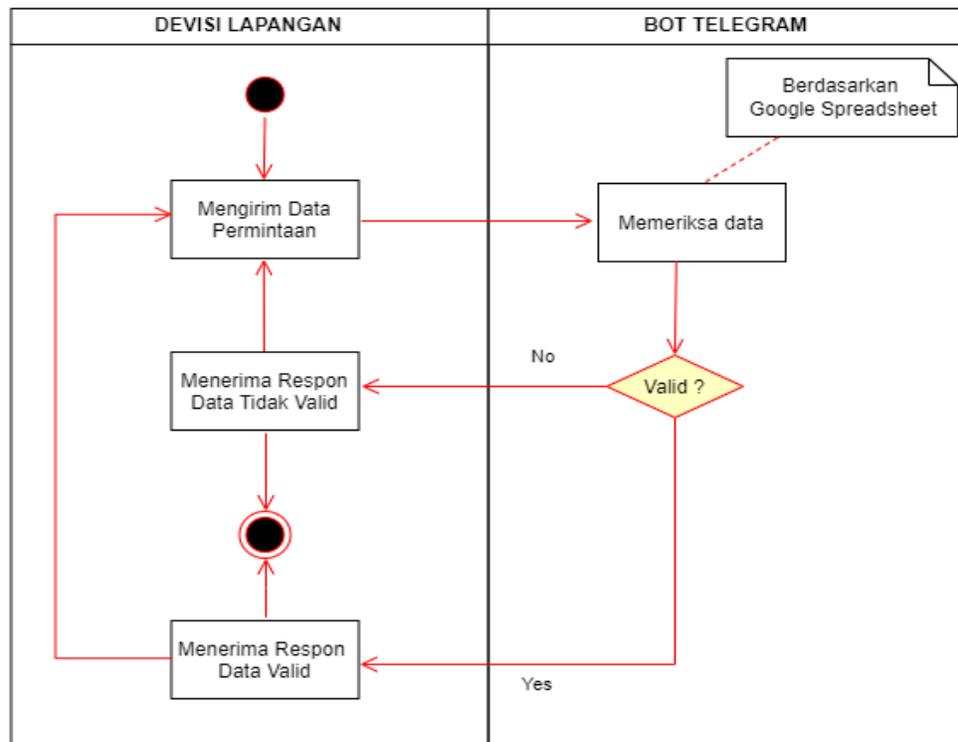
Gambar 4.1 Activity Diagram Sistem Lama

- Activity Diagram di mulai dari Devisi Lapangan mengirim data ke Devisi *Office*
- Devisi *Office* memeriksa data berdasarkan *database Google Spreadsheet* dan mengirimkan respon ke Devisi Lapangan, jika data tersebut valid maka respon yang dikirim ke Devisi lapangan berupa data yang diminta, jika data tersebut tidak valid maka respon yang di kirim adalah respon tidak valid atau data tidak ada di *database*

- Devisi Lapangan menerima respon dari Devisi *Office*.

## 2. Alur Sistem Baru

Alur sistem baru menggambarkan alur sistem yang diusulkan pada penelitian ini, berikut ini gambaran dari alur sistem yang baru.



Gambar 4.2 Activity Diagram Sistem Baru

- Activity Diagram di mulai dari Devisi Lapangan mengirim data ke Bot Telegram
- Dengan Algoritma yang ditanam Bot Telegram memeriksa data berdasarkan *database Google Spreadsheet* dan mengirimkan respon ke Devisi Lapangan, jika data tersebut valid maka respon yang dikirim berupa data yang diminta, jika data tersebut tidak valid maka respon yang di kirim adalah respon tidak valid atau data tidak ada di *database*

- Devisi Lapangan menerima respon dari Bot Telegram.

Alur kerja sistem antara sistem lama dengan sistem baru tidak banyak perubahan atau bisa dikatakan sama, yang membedakan adalah entitas dari sistem tersebut yaitu Devisi *Office* di ganti dengan Bot Telegram.

### 4.3. ALGORITMA SISTEM

Dalam penelitian ini terdapat algoritma yang ditanam di sistem. Algoritma ini di tanam pada bot telegram dan *google spreadsheet* yang akan di tampilkan sebagai berikut:

#### 1. Algoritma Pada Bot Telegram

Algoritma pada bot telegram berfungsi untuk mengirimkan perintah yang telah di masukkan ke *Google Spreadsheet*. Ini berupa link dan API yang di arahkan dari bot Telegram ke *Google Spreadsheet*.

#### 2. Algoritma pada *Google Spreadsheet*

Algoritma pada *google spreadsheet* dibagi menjadi beberapa kategori sebagai berikut:

##### a. Algoritma pada *Google Apps Script*

*Google Apps Script* adalah *platform* skrip yang dikembangkan oleh *Google* untuk pengembangan aplikasi ringan di *platform Google Workspace*. Bahasa skrip yang di gunakan adalah bahasa pemograman Java. Berikut ini gambaran dari algoritma yang ditanam pada *Google Apps Script*

- Fungsi *doPost(e)*

Fungsi ini dibuat untuk mengcapture, menganalisa dan membalas permintaan dari user melalui bot. Fungsi ini bisa dikatakan fungsi utama atau *Main Fuction*. Fungsi ini dapat dibagi menjadi beberapa formula IF sebagai berikut:

- 1) Jika yang di input adalah titik kordinat dengan metode “Send My Current Location” dan tidak mengirim teks. Formula ini berguna untuk memfilter perintah yang khusus dengan metode tersebut, kemudian data di tangkap dan di arahkan ke Sheet Find Near pada cell B2, dengan algoritma yang sudah di tanam pada Sheet Find Near akan mengkalkulasi data yang di input dan menampilkan data yang di inginkan jika memang data tersebut ada. Kemudian user yang mengirim permintaan akan di catat pada Sheet All Visitor. Berikut ini adalah *script* dari formula ini.

```

if (updates.message.text!=null) {
  var cordinate = geostring.latitude + "," + geostring.longitude;
  //kirm ke telegram
  var sheetx = SpreadsheetApp.openById(SheetID).getSheetByName('Find Near')
;

  //input to search
  sheetx.getRange(2, 2).setValue(cordinate);
  var xyz = sheetx.getRange(2, 17).getValue();

  if (xyz == "1") {
    sendText(updates.message.chat.id, notif);
    sendText(updates.message.chat.id, bantuan);
  }

  else if (xyz >= 10) {
    //perulangan
    for (var row = 2; row <= 10; row++) {
      variables += sheetx.getRange(row, 11).getValue() + " | " + sheetx.
getRange(row, 13).getValue() + " | " + sheetx.getRange(row, 14).getValue().
toFixed(0) + "M | " + sheetx.getRange(row, 21).getValue() + " | " + sheetx.
getRange(row, 12).getValue() + "\n";
    }

    //Send Message
    sendText(updates.message.chat.id, label + variables);
    sendText(updates.message.chat.id, bantuan);

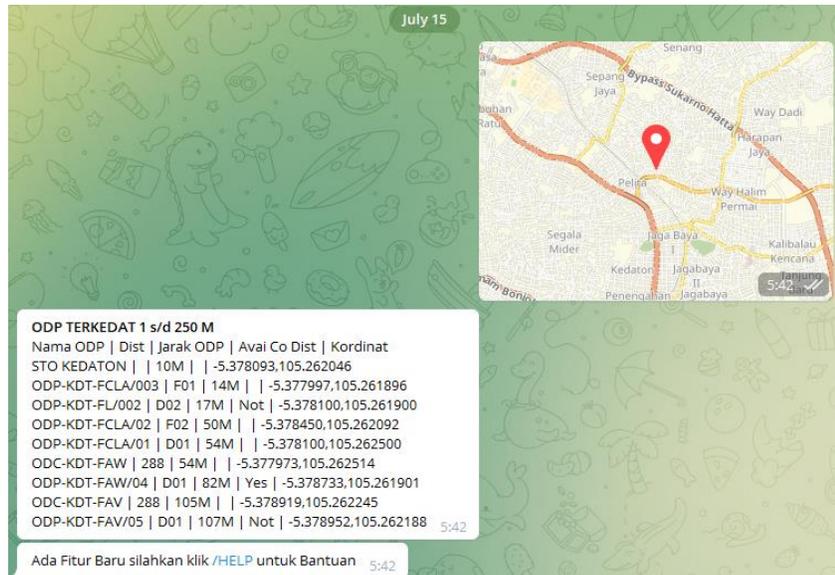
    //input to sheet Visitor
    var sheet2 = SpreadsheetApp.openById(SheetID).getSheetByName('All
Visitor');
    var last = sheet2.getRange('A1').getDataRegion().getLastRow();
    sheet2.getRange(last + 1, 1).setValue(new Date());
    sheet2.getRange(last + 1, 2).setValue(updates.message.chat.id);
    sheet2.getRange(last + 1, 3).setValue(updates.message.chat.first_name
+ ' ' + updates.message.chat.last_name);
    sheet2.getRange(last + 1, 4).setValue(updates.message.chat.username);
    sheet2.getRange(last + 1, 5).setValue(cordinate);
  }

  else {

```

Gambar 4.3 Script IF dengan metode Send My Current Location

Adapun contoh respon yang ditampilkan dari formula ini adalah sebagai berikut:



Gambar 4.4 Respon valid dengan metode Send My Current Location



Gambar 4.5 Respon not valid dengan metode Send My Current Location

- 2) Jika yang di input adalah karakter /START, formula ini berguna untuk menampilkan deskripsi dari Bot Telegram, berikut ini adalah *script* yang terdapat pada formula ini

```

else if (updates.message.text.toUpperCase() == "/START") {
    //kirim ke bot telegram ketika ada perintah /start
    sendText(updates.message.chat.id, "Hello Guys, Selamat Datang Di BOT
@LabelingODPBOT, BOT ini digunain buat mencari titik ODP atau ODC di Lampung,
Jika ingin mengetahui Perintah di BOT ini silahkan klik /HELP. Semoga BOT ini
dapat membatu temen-temen di Lapangan. TerimaKasih. Supported By TIDC Team
Lampung (@RudyTabootie, @AROMANIKMAT, @redhoekap, @Nirapeb, @raniratna,
@R_Gutama, @khafidns, @SabarTakNgopiSekJal, @nndosptraa). Developed by Khafid
Nursalim");
}

```

Gambar 4.6 Script IF dengan karakter /START

Adapun contoh respon yang ditampilkan dari formula ini adalah sebagai berikut:



Gambar 4.7 Respon Bot dengan karakter /START

- 3) Jika yang di input adalah karakter /HELP, formula ini berguna untuk menampilkan prosedur penggunaan aplikasi. Berikut ini *script* yang terdapat dalam formula ini:

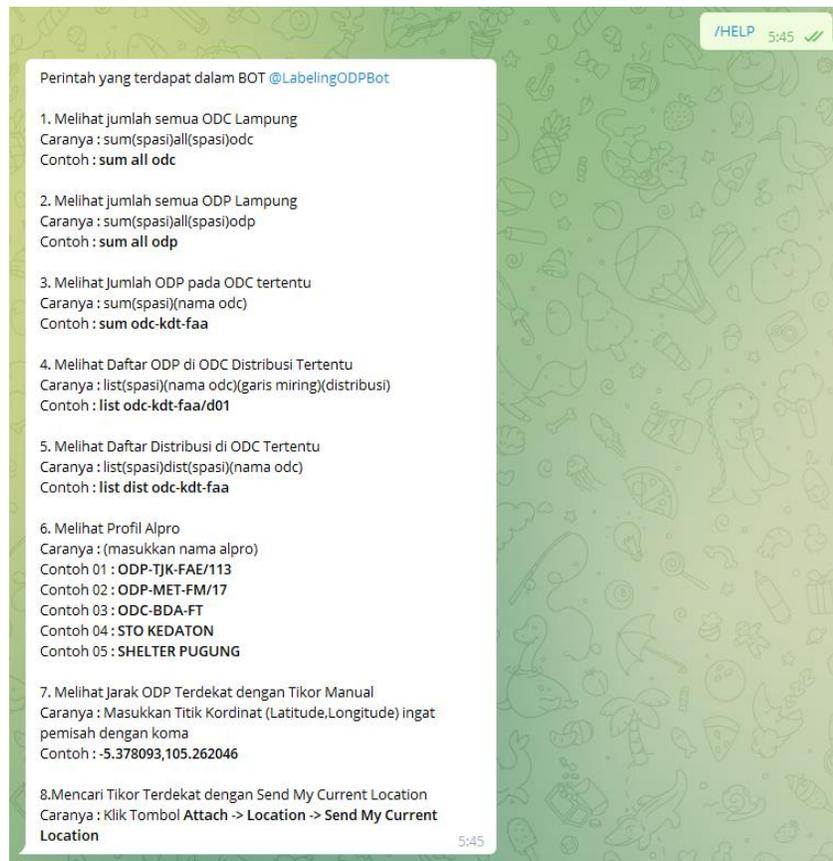
```

else if (updates.message.text.toUpperCase() == "/HELP") {
    //kirim ke bot telegram ketika ada perintah /start
    var helps = "Perintah yang terdapat dalam BOT @LabelingODPBot\n\n1. Melihat
jumlah semua ODC Lampung\nCaranya : sum(spasi)all(spasi)odc\nContoh : <b>sum all
odc</b>\n\n2. Melihat jumlah semua ODP Lampung\nCaranya : sum(spasi)all(spasi)
odp\nContoh : <b>sum all odp</b>\n\n3. Melihat Jumlah ODP pada ODC
tertentu\nCaranya : sum(spasi)(nama odc)\nContoh : <b>sum odc-kdt-faa</b>\n\n4.
Melihat Daftar ODP di ODC Distribusi Tertentu\nCaranya : list(spasi)(nama odc)
(garis miring)(distribusi)\nContoh : <b>list odc-kdt-faa/d01</b>\n\n5. Melihat
Daftar Distribusi di ODC Tertentu\nCaranya : list(spasi)dist(spasi)(nama odc)
\nContoh : <b>list dist odc-kdt-faa</b>\n\n6. Melihat Profil Alpro\nCaranya :
(masukkan nama alpro)\nContoh 01 : <b>ODP-TJK-FAE/113</b>\nContoh 02 :
<b>ODP-MET-FM/17</b>\nContoh 03 : <b>ODC-BDA-FT</b>\nContoh 04 : <b>STO KEDATON</
b>\nContoh 05 : <b>SHELTER PUGUNG</b>\n\n7. Melihat Jarak ODP Terdekat dengan
Tikor Manual\nCaranya : Masukkan Titik Kordinat (Latitude,Longitude) ingat
pemisah dengan koma\nContoh : <b>-5.378093,105.262046</b> \n\n8.Mencari Tikor
Terdekat dengan Send My Current Location\nCaranya : Klik Tombol <b>Attach ->
Location -> Send My Current Location</b>";
    sendText(updates.message.chat.id, helps)
}

```

Gambar 4.8 Script IF dengan karakter /HELP

Adapun contoh respon yang ditampilkan dari formula ini adalah sebagai berikut:



Gambar 4.9 Respon Bot dengan karakter /HELP

- 4) Jika yang di input adalah karakter SUM ALL ODP, formula ini berguna untuk menampilkan jumlah seluruh ODP yang tersimpan dalam *database*. Berikut ini *script* yang terdapat dalam formula ini:

```

else if (updates.message.text.toUpperCase() == "SUM ALL ODP") {
//kirim ke bot telegram ketika ada perintah /start
var sheetxx = SpreadsheetApp.openById(SheetID).getSheetByName('Jumlah ODP|ODC');
var jml = "Jumlah ODP seluruh Lampung adalah "+sheetxx.getRange(1, 8).getValue();
sendText(updates.message.chat.id, jml);
sendText(updates.message.chat.id, bantuan);

//input visitor
var sheet2 = SpreadsheetApp.openById(SheetID).getSheetByName('All Visitor');
var last = sheet2.getRange('A1').getDataRegion().getLastRow();
sheet2.getRange(last + 1, 1).setValue(new Date());
sheet2.getRange(last + 1, 2).setValue(updates.message.chat.id);
sheet2.getRange(last + 1, 3).setValue(updates.message.chat.first_name + ' ' + updates.
message.chat.last_name);
sheet2.getRange(last + 1, 4).setValue(updates.message.chat.username);
sheet2.getRange(last + 1, 5).setValue(updates.message.text);
}

```

Gambar 4.10 Script IF dengan karakter SUM ALL ODP

Adapun contoh respon yang ditampilkan dari formula ini adalah sebagai berikut:



Gambar 4.11 Respon Bot dengan karakter SUM ALL ODP

- 5) Jika yang di input adalah karakter SUM ALL ODC, formula ini berguna untuk menampilkan jumlah seluruh ODC yang tersimpan dalam *database*. Berikut ini *script* yang terdapat dalam formula ini:

```

else if (updates.message.text.toUpperCase() == "SUM ALL ODC") {
//kirim ke bot telegram ketika ada perintah /start
var sheetxx = SpreadsheetApp.openById(SheetID).getSheetByName('Jumlah ODP|ODC');
var jml = "Jumlah ODC seluruh Lampung adalah "+sheetxx.getRange(1, 9).getValue();
sendText(updates.message.chat.id, jml);
sendText(updates.message.chat.id, bantuan);

//input visitor
var sheet2 = SpreadsheetApp.openById(SheetID).getSheetByName('All Visitor');
var last = sheet2.getRange('A1').getDataRegion().getLastRow();
sheet2.getRange(last + 1, 1).setValue(new Date());
sheet2.getRange(last + 1, 2).setValue(updates.message.chat.id);
sheet2.getRange(last + 1, 3).setValue(updates.message.chat.first_name + ' ' + updates.
message.chat.last_name);
sheet2.getRange(last + 1, 4).setValue(updates.message.chat.username);
sheet2.getRange(last + 1, 5).setValue(updates.message.text);
}

```

Gambar 4.12 Script IF dengan karakter SUM ALL ODC

Adapun contoh respon yang ditampilkan dari formula ini adalah sebagai berikut:



Gambar 4.11 Respon Bot dengan karakter SUM ALL ODC

#### 6) Jika yang di input adalah karakter LIST ODC

```

else if (updates.message.text.toUpperCase().slice(0, 8) == "LIST ODC") {
    //kirim ke bot telegram ketika ada perintah ODC

    var sheetxx = SpreadsheetApp.openById(SheetID).getSheetByName('List ODP Dist');
    sheetxx.getRange(1, 2).setValue(updates.message.text.toUpperCase().slice(5, 20));
    var veri = sheetxx.getRange(1, 4).getValue();

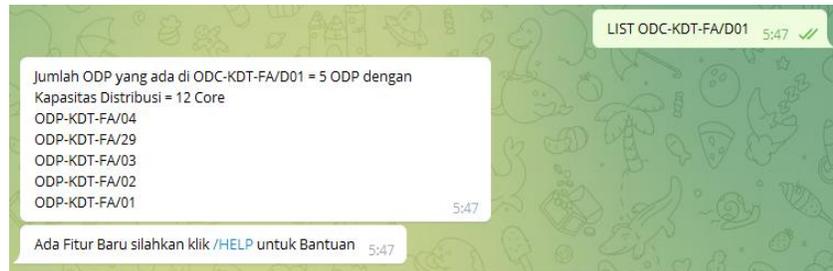
    //input visitor
    var sheet2 = SpreadsheetApp.openById(SheetID).getSheetByName('All Visitor');
    var last = sheet2.getRange('A1').getDataRegion().getLastRow();
    sheet2.getRange(last + 1, 1).setValue(new Date());
    sheet2.getRange(last + 1, 2).setValue(updates.message.chat.id);
    sheet2.getRange(last + 1, 3).setValue(updates.message.chat.first_name + ' ' + updates.
message.chat.last_name);
    sheet2.getRange(last + 1, 4).setValue(updates.message.chat.username);
    sheet2.getRange(last + 1, 5).setValue(updates.message.text);

    if(veri == "2"){
        sendText(updates.message.chat.id, updates.message.text.toUpperCase().slice(5, 20)+"
Tidak ada di sistem" );
        sendText(updates.message.chat.id, bantuan);
    }
    else {
        var www = "Jumlah ODP yang ada di " + updates.message.text.toUpperCase().slice(5,
20) + " = " + sheetxx.getRange(1, 5).getValue()+" ODP dengan Kapasitas Distribusi = "+
sheetxx.getRange(1, 9).getValue()+" Core\n";
        for (var row = 3; row <= veri; row++) {
            variables += sheetxx.getRange(row, 1).getValue()+"\n";
        }
        sendText(updates.message.chat.id, www + variables);
        sendText(updates.message.chat.id, bantuan);
    }
}

```

Gambar 4.12 Script IF dengan karakter LIST ODC

Adapun contoh respon yang ditampilkan dari formula ini adalah sebagai berikut:



Gambar 4.13 Respon Bot dengan karakter LIST ODC

## 7) Jika yang di input adalah karakter LIST DIS

```

else if (updates.message.text.toUpperCase().slice(0, 8) == "LIST DIS") {
    //kirim ke bot telegram ketika ada perintah ODC

    var sheetxx = SpreadsheetApp.openById(SheetID).getSheetByName('Dist Search');
    sheetxx.getRange(1, 2).setValue(updates.message.text.toUpperCase().slice(10, 21) + "/");
    var veri = sheetxx.getRange(1, 4).getValue();

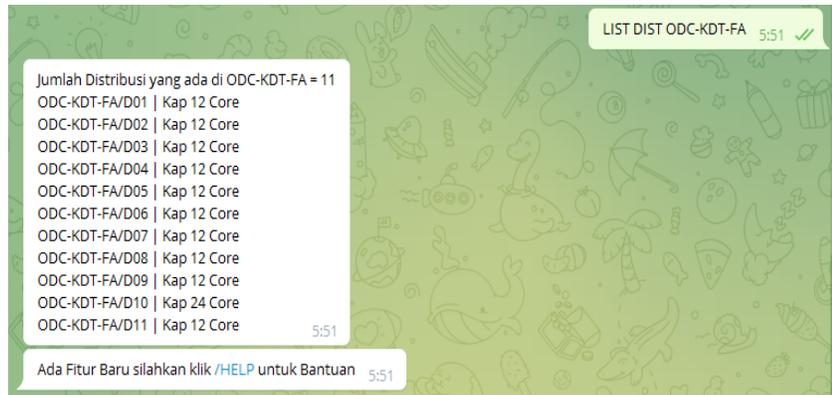
    //input visitor
    var sheet2 = SpreadsheetApp.openById(SheetID).getSheetByName('All Visitor');
    var last = sheet2.getRange('A1').getDataRegion().getLastRow();
    sheet2.getRange(last + 1, 1).setValue(new Date());
    sheet2.getRange(last + 1, 2).setValue(updates.message.chat.id);
    sheet2.getRange(last + 1, 3).setValue(updates.message.chat.first_name + ' ' + updates.
message.chat.last_name);
    sheet2.getRange(last + 1, 4).setValue(updates.message.chat.username);
    sheet2.getRange(last + 1, 5).setValue(updates.message.text);

    if(veri == "0"){
        sendText(updates.message.chat.id, updates.message.text.toUpperCase().slice(10, 21) +
Tidak ada di sistem" );
        sendText(updates.message.chat.id, bantuan);
    }
    else {
        var www = "Jumlah Distribusi yang ada di " + updates.message.text.toUpperCase().slice
(10, 21) + " = " + sheetxx.getRange(1, 5).getValue() + "\n";
        for (var row = 2; row <= veri; row++) {
            variables += sheetxx.getRange(row, 1).getValue() + " | Kap " + sheetxx.getRange(row,
2).getValue() + " Core\n";
        }
        sendText(updates.message.chat.id, www + variables);
        sendText(updates.message.chat.id, bantuan);
    }
}

```

Gambar 4.14 Script IF dengan karakter LIST DIS

Adapun contoh respon yang ditampilkan dari formula ini adalah sebagai berikut:



Gambar 4.15 Respon Bot dengan karakter LIST DIS

- Fungsi `getRows`

Fungsi ini dibuat untuk mengambil data dari *Google Spreadsheet* dimasukkan kedalam penyimpanan sementara guna untuk mengkalkulasi dan mendapatkan informasi tentang data-data yang ada.

```
function getRows() {
  //allocate range gsheet
  var rangeName = 'Sheet1!A2:Q';
  var rows = Sheets.Spreadsheets.Values.get(SheetID, rangeName).values;
  return rows;
}
```

Gambar 4.16 *Script* fungsi `getRows`

- Fungsi `searchOdc`

Fungsi ini dibuat untuk mencari data ODC berdasarkan fungsi `getRows` yang telah mempunyai data yang di simpan dalam tempat penyimpanan sementara. Data yang diambil adalah data Titik Kordinat, Kapasitas ODC, Tgl Golive, Nama Project dan Type Project. Berikut ini *script* dari Fungsi `searchOdc`

```

function searchOdc(idOdp) {
    //variable find data with uppercase
    var data = idOdp.toUpperCase();
    var dataOdp = getRows();

    //find data
    for (var row = 0; row < dataOdp.length; row++) {
        if (dataOdp[row][0] == data) {
            kordinat = dataOdp[row][1];
            return "<u><b>PROFIL ODC</b></u>\n<b>Nama ODC : </b>" + dataOdp[row][0] + "\n" +
                "<b>Titik Kordinat : </b>" + dataOdp[row][1] + "\n" +
                "<b>Kapasitas ODC : </b>" + dataOdp[row][2] + "\n" +
                "<b>Tgl Golive : </b>" + dataOdp[row][3] + "\n" +
                "<b>Nama Project : </b>" + dataOdp[row][5] + "\n" +
                "<b>Type Project : </b>" + dataOdp[row][4] ;
        }
    }
    return "Sorry ya Guys, data gak ketemu, kemungkinan data belum di Input atau kalian
    typo, silahkan cek lagi penulisannya, Terima Kasih";
}

```

Gambar 4.17 Script fungsi searchOdc

- Fungsi searchOdpById1

Fungsi ini dibuat untuk mencari data ODP berdasarkan fungsi getRows yang telah mempunyai data yang di simpan dalam tempat penyimpanan sementara. Data yang diambil adalah data Nama ODP, Distribusi, Titik Kordinat, Tgl Golive, Nama Project Type Project, Core Distribusi, Core Feeder, IP GPON, Modul GPON, Port GPON, Lokasi GPON, Merek GPON, ID SW dan Mitra Pembangun

```

function searchOdpById1(idOdp) {
    //variable find data with uppercase
    var data = idOdp.toUpperCase();
    var dataOdp = getRows();

    //find data
    for (var row = 0; row < dataOdp.length; row++) {
        if (dataOdp[row][0] == data) {
            kordinat = dataOdp[row][1];
            return "<u><b>PROFIL ODP</b></u>\n<b>Nama ODP : </b>" + dataOdp[row][0] + "\n" +
                "<b>Distribusi : </b>" + dataOdp[row][2] + "\n" +
                "<b>Titik Kordinat : </b>" + dataOdp[row][1] + "\n" +
                "<b>Tgl Golive : </b>" + dataOdp[row][3] + "\n" +
                "<b>Nama Project : </b>" + dataOdp[row][5] + "\n" +
                "<b>Type Project : </b>" + dataOdp[row][4] + "\n" +
                "<b>Core Distribusi : </b>" + dataOdp[row][8] + "\n" +
                "<b>Core Feeder : </b>" + dataOdp[row][9] + "\n" +
                "<b>IP GPON : </b>" + dataOdp[row][10] + "\n" +
                "<b>Modul GPON : </b>" + dataOdp[row][11] + "\n" +
                "<b>Port GPON : </b>" + dataOdp[row][12] + "\n" +
                "<b>Lokasi GPON : </b>" + dataOdp[row][16] + "\n" +
                "<b>Merek GPON : </b>" + dataOdp[row][15] + "\n" +
                "<b>ID SW : </b>" + dataOdp[row][14] + "\n" +
                "<b>Mitra Pembangun : </b>" + dataOdp[row][13];
        }
    }
    return "Sorry ya Guys, data gak ketemu, kemungkinan data belum di Input atau kalian
    typo, silahkan cek lagi penulisannya, Terima Kasih";
}

```

Gambar 4.18 Script fungsi searchOdpById1

#### - Fungsi SentText

Fungsi ini dibuat sebagai fungsi untuk mengirimkan data ke bot Telegram

```

function sendText(chatid, text, replymarkup) {
    var data = {
        method: "post",
        payload: {
            method: "sendMessage",
            chat_id: String(chatid),
            text: text,
            parse_mode: "HTML",
            reply_markup: JSON.stringify(replymarkup)
        }
    };
    UrlFetchApp.fetch('https://api.telegram.org/bot' + token + '/', data);
}

```

Gambar 4.19 Script fungsi sendText

b. Algoritma pada Sheet Sheet1

Algoritma yang di tanam pada Sheet Sheet1 adala berupa formula excel yang bertujuan untuk mendapatkan informasi yang lebih berguna dengan inputan yang sedikit. Berikut ini formula yang ditanam

- Pada *field* CEK DUPLIKASI

```
=countif($A:$A,A2)
```

*Field* ini berfungsi untuk memeriksa karakter yang duplikasi berdasarkan *field* ODP/ODC/STO. Jika terdapat duplikasi *field* ini akan menghitung jumlah kata yang berulang, jika tidak terdapat duplikasi *field* ini akan menunjukkan angka 1.

- Pada *field* STO

```
=if(A2="", "", MID(A2, FIND("-", A2) +  
1, FIND(CHAR(160), SUBSTITUTE(A2, "-", CHAR(160), 2)) - 1 -  
(FIND("-", A2))))
```

*Field* ini berfungsi untuk menampilkan STO yang terdapat pada *field* ODP/ODC/STO dengan memeriksa terlebih dahulu isi dari *field* ODP/ODC/STO, jika tidak terdapat isi dalam field, *field* ini tidak menunjukkan isi, jika terdapat isi *field* ini akan meneruskan algoritmanya, yaitu mengambil karakter tengah yang diapit oleh karakter “-“

- Pada *field* MEREK OLT

```
=IF(K2="", "", VLOOKUP(K2, 'Info Olt'!$A$2:$D$113, 4, 0))
```

*Field* ini berfungsi untuk menampilkan merek olt berdasarkan *field* IP yang disinkronkan dengan sheet Info Olt.

- Pada *field* POWER OLT

```
=IF(K2="", "", VLOOKUP(K2, 'Info Olt'!$A$2:$B$113, 2, FALSE))
```

*Field* ini berfungsi untuk menampilkan STO sebagai sumber dari odp berdasarkan *field* IP yang disinkronkan dengan sheet Info Olt.

- Pada *field* FORMULASI

```
=if(mid(A2,11,1)="/", "ODC-  
"&MID(A2,5,6)&"/"&C2, if(mid(A2,12,1)="/", "ODC-  
"&MID(A2,5,7)&"/"&C2, ""))
```

*Field* ini berfungsi untuk menampilkan catuan ODC dan Distribusi berdasarkan *field* ODP/ODC/STO, jika tidak terdapat isi dalam filed, *field* ini tidak menunjukkan isi, jika terdapat isi *field* ini akan meneruskan algoritmanya, yaitu menambah karakter “ODC-“ dan mengambil karakter setelah karakter “/” dan ditambah dengan karakter yang ada di *field* “DIS”

- Pada *field* LATITUDE & LONGITUDE.

```
=IF(B2="", "", SPLIT(TRIM(B2), ","))
```

*Field* ini berfungsi untuk menampilkan dan memisahkan kordinat berdasarkan *field* “TIKOR” dengan tanda koma “,” sebagai pemisah.

- Pada *field* CATUAN ODC

```
=if(mid(A2,11,1)="/", "ODC-  
"&MID(A2,5,6), if(mid(A2,12,1)="/", "ODC-"&MID(A2,5,7), ""))
```

*Field* ini berfungsi untuk menampilkan ODC sebagai sumber ODP dengan menambahkan karakter “ODC-“ dan mengambil karakter setelah karakter “-“ pertama pada *field* ODP/ODC/STO.

c. Algoritma pada Sheet All Visitor

Algoritma yang di tanam pada Sheet All Visitor ini bertujuan untuk melihat jumlah pengguna keseluruhan dari bot telegram yang terdapat pada penelitian ini. Berikut ini formula yang di tanam

- Pada seluruh *field* VERIFY

=IF(LEFT(E2,1)="-",1, countif(Sheet1!\$A:\$A,E2))

*Field* ini berfungsi untuk memeriksa perintah yang dapat di layani oleh sistem, jika perintah tersebut dapat di layani maka akan menampilkan 1, tetapi jika tidak dapat di layani akan menampilkan 0.

- Pada *field* TOTAL REQUEST

=COUNTA(E:E)-1

*Field* ini berfungsi untuk menghitung seluruh permintaan yang di terima oleh bot dengan menghitung *field* “Username” yang terisi.

- Pada *field* SUCCESS REQUEST

=SUM(G:G)

*Field* ini berfungsi untuk menghitung jumlah permintaan yang dapat di layani dengan menjumlahkan *field* “Verify”

- Pada *field* TOTAL VISITOR

=COUNTUNIQUE(D:D)

*Field* ini berfungsi untuk menghitung jumlah pengguna yang mengakses bot ini. Rumus dari *field* ini adalah menghitung pengguna yang unik pada *field* “Username”

- Pada *field* NO, VISITOR, COUNT

```
=SORT(M2:N336,N2:N336,false)
```

*Field* ini berfungsi untuk melihat user yang sering menggunakan bot. rumus dari *field* ini adalah mengurutkan username berdasarkan jumlah permintaan terbanyak.

d. Algoritma pada Sheet One Day History

Algoritma yang di tanam pada Sheet One Day History ini bertujuan untuk melihat jumlah pengguna per hari yang melakukan permintaan pada Bot telegram. Berikut ini formula yang di tanam

- Pada *Field* TODAY

```
=TODAY()
```

*Field* ini berfungsi untuk menampilkan tanggal hari ini.

- Filed yang lain

Untuk *field* yang lain berfungsi sama dengan Algoritma yang terdapat pada Sheet All Visitor, perbedaannya terdapat pada perhitungan. Jika pada Sheet All Visitor perhitungannya adalah total keseluruhan sedangkan pada Sheet One Day History perhitungannya adalah perhari.

e. Algoritma pada Sheet Find Near

Algoritma yang ditanam pada Sheet Find Near tergolong kompleks karena terdapat beberapa algoritma. Algoritma yang ditanam pada Sheet ini bertujuan untuk melihat jarak ODP atau alpro (alat produksi) terdekat dari titik kordinat yang di masukkan. Berikut ini formula yang di tanam

- Pada *field* ODP, KORDINAT, DISTRIBUSI

```
=FILTER(Sheet1!A:C,(Sheet1!T:T>=C5)*(Sheet1!T:T<=C4),(Sheet1!U:U>=D5)*(Sheet1!U:U<=D4))
```

*Field* ini berfungsi untuk menampilkan daftar yang mengandung kordinat berdasarkan *field* input max dan *field* input min

- Pada *field* DISTANCE

```
=IF(I2="", "", 2 * 6371000 * ASIN(SQRT((SIN((I2*(3.14159/180))- $C$2*(3.14159/180))/2))^2+COS(I2*(3.14159/180))*COS($C$2*(3.14159/180))*SIN(((J2*(3.14159/180))- $D$2*(3.14159/180))/2))^2))
```

*Field* ini berfungsi untuk mengkalkulasi jarak antara tikor yang dimasukkan dengan list yang ditampilkan

- *Field* ODP (SORT)

```
=if(F2="", "", sort(E2:H, H2:H, TRUE))
```

*Field* ini berfungsi untuk mengurutkan berdasarkan jarak terdekat

f. Algoritma pada Sheet Skema ODP

Algoritma yang ditanam pada Sheet Find Skema ODP tergolong kompleks karena terdapat beberapa algoritma. Algoritma yang ditanam pada Sheet ini bertujuan untuk melihat layout dari suatu distribusi

tertentu berikut kapasitas distribusi dan jumlah odp yang terdapat dalamnya. Berikut ini formula yang di tanam

- Pada *field* CORE FEE USED

```
=IF(B2="", "", ROUNDUP(SUMIF($L$2:$N$1864,B2,$M$2:$M$1864)/4,0))
```

*Field* ini berfungsi untuk menghitung jumlah core yang digunakan dalam suatu feeder berdasarkan pebulatan dari kalkulasi *field* CORE USED dibagi menjadi 4

- Pada *field* CORE FEE IDLE

```
=IF(C2="", "", C2-(D2+F2))
```

*Field* ini berfungsi untuk menghitung jumlah core yang tersedia dalam suatu feeder berdasarkan kalkulasi *field* KAP FEE diambil dengan *field* CORE FEE USED

- Pada *field* KET FEEDER

```
=IF(E2="", "", IF(E2=0, "HABIS", IF(E2<=3, "KRITIS", "IDLE")))
```

*Field* ini berfungsi untuk melihat status apakah suatu feeder masih dapat digunakan atau dapat di pakai untuk pembangunan, parameter yang ada berupa keterangan HABIS jika core habis, KRITIS jika core tersisa kurang dari 3 dan IDLE jika core lebih dari 3.

- Pada *field* NAMA DISTRIBUSI

```
=IF(B2="", "", "DS-"&MID(A2,5,3)&"-FE-"&MID(B2,8,10)&"/"&RIGHT(A2,2))
```

*Field* ini berfungsi untuk memberi nama pada suatu distribusi tertentu berdasarkan *field* NAMA FEEDER dan *field* ODC DAN DIST

- Pada *field* CORE USED

=COUNTA(T2:AQ2)

*Field* ini berfungsi untuk mengkalkulasi jumlah ODP yang terdapat pada Distribusi tertentu

- Pada *field* CORE IDLE

=IF(K2="", "", K2-(M2+O2))

*Field* ini berfungsi untuk mengkalkulasi jumlah core distribusi yang tersisa dengan berdasarkan *field* CORE USED diambil *field* KAP DIST

- Pada *field* KET DIS

=IF(N2="", "", IF(N2=0, "HABIS", IF(N2<=3, "KRITIS", "IDLE")))

*Field* ini berfungsi untuk melihat status apakah suatu distribusi masih dapat digunakan atau dapat di pakai untuk pembangunan, parameter yang ada berupa keterangan HABIS jika core habis, KRITIS jika core tersisa kurang dari 3 dan IDLE jika core lebih dari 3.

- Pada *field* ODP1, ODP2 dst

=TRANPOSE(filter(Sheet1!S:S, Sheet1!R:R=A2))

*Field* ini berfungsi untuk menampilkan daftar ODP pada Distribusi Tertentu secara Horizontal berdasarkan *database* yang telah di Input pada Sheet Sheet1

g. Algoritma pada Sheet Jumlah ODP|ODC

Algoritma yang di tanam pada Sheet Jumlah ODP|ODC ini bertujuan untuk melihat jumlah ODP yang terdapat pada ODC tertentu dan untuk

melihat jumlah keseluruhan ODP yang terdapat dalam *database*. Berikut ini formula yang di tanam

- Pada *field* CATUAN ODC

```
=unique(filter(Sheet1!Z2:Z,Sheet1!Z2:Z<>""))
```

*Field* ini berfungsi untuk menampilkan daftar ODC berdasarkan Sheet Sheet1

- Pada *field* JUMLAH ODP

```
=IF(A2="", "", COUNTIF(Sheet1!Z:Z,A2))
```

*Field* ini berfungsi untuk menampilkan jumlah ODC berdasarkan *field* CATUAN ODC

- Pada cell H1

```
=sum(B:B)
```

Cell ini berfungsi untuk mengkalkulasi jumlah keseluruhan odp yang terdapat dalam *database*

#### h. Algoritma pada Sheet List ODP Dist

Algoritma yang di tanam pada Sheet List ODP Dist ini bertujuan untuk melihat daftar ODP yang terdapat pada distribusi tertentu. Berikut ini formula yang di tanam

- Pada *field* NAMA ODP

```
=IFERROR(if(B1="", "", FILTER(Sheet1!A:A,Sheet1!R:R=B1)), "")
```

*Field* ini berfungsi untuk menampilkan daftar ODP tertentu berdasarkan keyword yang terdapat pada cell B2 yang menyesuaikan dengan Sheet Sheet1

- Pada *field* CORE IDLE

=VLOOKUP(B1, 'Skema ODP'!\$A:\$N, 14, FALSE)

*Field* ini berfungsi untuk menampilkan core idle berdasarkan keyword yang terdapat pada cell B2 yang menyesuaikan dengan Sheet Skema ODP

- Pada *field* CORE BROKEN

=VLOOKUP(B1, 'Skema ODP'!\$A:\$O, 15, FALSE)

*Field* ini berfungsi untuk menampilkan core broken berdasarkan keyword yang terdapat pada cell B2 yang menyesuaikan dengan Sheet Skema ODP

- Pada *field* CAP CORE

=VLOOKUP(B1, 'Skema ODP'!\$A:\$K, 11, FALSE)

*Field* ini berfungsi untuk menampilkan kapasitas core berdasarkan keyword yang terdapat pada cell B2 yang menyesuaikan dengan Sheet Skema ODP

#### 4.4. ANALISIS PERBANDINGAN SISTEM LAMA DENGAN SISTEM BARU

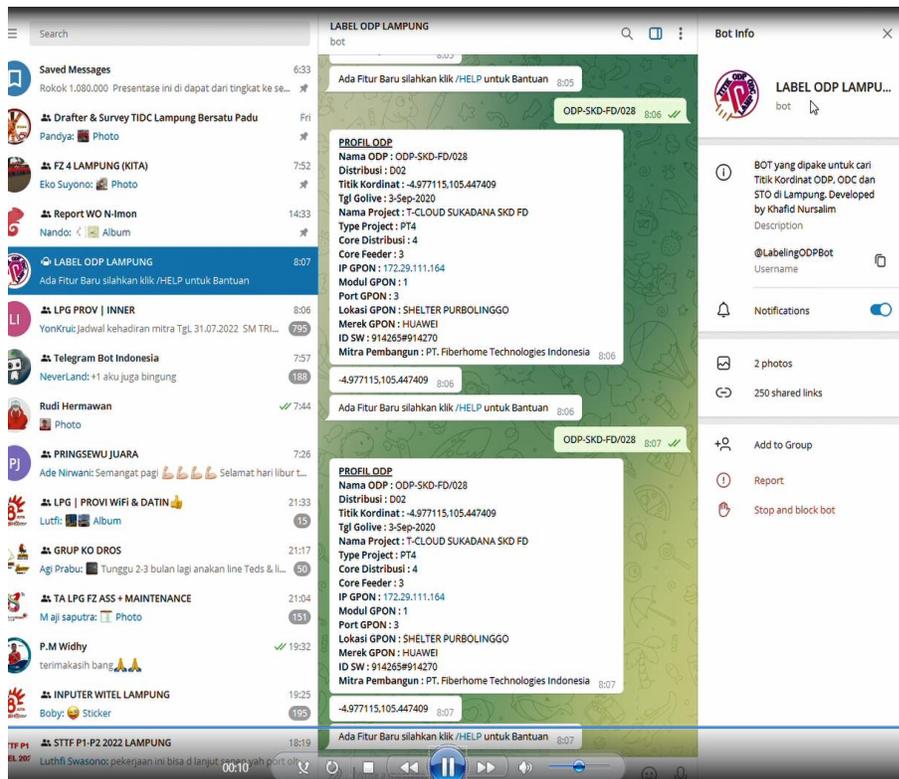
Analisis perbandingan sistem ini bertujuan untuk melihat tingkat kemanfaatan dari sistem baru dengan sistem lama. Adapun perbandingan sistem lama dengan sistem yang baru dapat dilihat pada tabel berikut:

NO	INDIKATOR	SISTEM LAMA	SISTEM BARU
1	Lama Waktu Respon	22 detik	10 detik
2	Waktu Standby	8 Jam	24 Jam
3	Penyajian Data	50 %	75 %
4	Input Karakter Sensitif Order	60 %	90%
5	Campur Tangan Manusia	100 %	50%

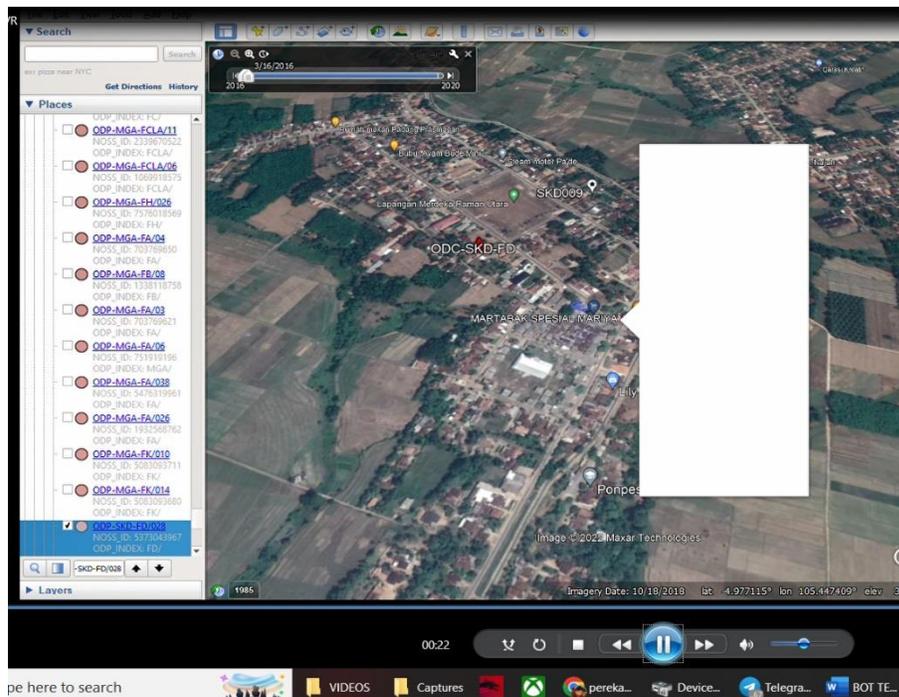
Tabel 4.1 Perbandingan Sistem

##### 1. Lama Waktu Respon

Pada indikator ini dilakukan percobaan yang dapat di simulasikan yaitu dengan melihat respon sistem terhadap perintah. Sistem lama memiliki respon 23 detik sedangkan sistem baru memiliki respon 10 detik. Sehingga untuk indikator ini sistem baru lebih unggul dibandingkan sistem yang lama dimana semakin kecil waktu respon akan semakin baik.



Gambar 4.20 Waktu Respon Sistem Baru



Gambar 4.21 Waktu Respon Sistem Lama

## 2. Waktu Standby

Indikator ini menandakan waktu sistem untuk dapat standby atau idle dalam satu hari. Pada sistem lama waktu standby dilihat dari jam kerja karyawan yaitu dari jam 08.00 s/d 17.00 dipotong waktu istirahat selama 1 jam, sedangkan untuk sistem yang baru waktu standby sistem adalah 24 jam, sehingga pada indikator ini sistem baru lebih unggul dibandingkan sistem yang lama dimana waktu standby sistem yang lebih lama akan semakin baik.

## 3. Penyajian data

Indikator ini mengindikasikan penyajian data sistem ke user. Kelengkapan penyajian data sistem lama sangat terbatas yaitu 50% dilihat dari informasi yang di tampilkan dalam sistem. Adapun data yang di tampilkan berupa nama odp, index odp, dan tikor odp. Sedangkan dalam sistem yang baru data yang di tampilkan berupa nama odp, distribusi, kordinat, tanggal golive, type project, core distribusi, core feeder, ip gpon, modul gpon, port gpon, lokasi gpon, merek gpon, ID SW, dan Mitra Pembangun.

## 4. Input Karakter Sensitif Order

Indikator ini menandakan tingkat kesensitifitasan sistem terhadap perintah atau order yang dimasukkan. Pada sistem yang lama kesensitifitasannya dapat ditolerir sebesar 60% karena yang membaca dan menterjemahkan adalah manusia sebagai contoh permintaan dari user tidak harus standar yang penting dapat di logika dengan manusia sedangkan tingkat ke sensitifan dari

sistem baru sebesar 90 % didapat dari permintaan dari user harus standar tetapi sistem dapat mentolerir huruf capital dan non capital. Sehingga untuk indikator ini sistem lama lebih unggul dibandingkan sistem yang baru dimana semakin kecil presentase indikator ini akan semakin baik.

#### 5. Campur Tangan Manusia

Indikator ini menandakan tingkat keterlibatan manusia dalam sistem. Pada sistem lama campur tangan manusia sangat dominan sekitar 100 % dibandingkan dengan sistem yang baru, sistem yang baru campur tangan manusia sekitar 50% karena dibantu dengan penggunaan algoritma yang di tanam pada sistem

## **BAB V**

### **SIMPULAN DAN SARAN**

#### **5.1. SIMPULAN**

Adapun simpulan dari uraian dan pembahasan pada Bab I sampai dengan Bab IV adalah sebagai berikut :

1. Pada penelitian ini aplikasi yang dibuat berupa BOT (Robot) pada aplikasi telegram dan dikombinasikan dengan *Google Spreadsheet*.
2. Algoritma ditanam pada *Google Spreadsheet* dengan beberapa formula sederhana yang di kombinasikan sehingga dapat memperoleh informasi yang di inginkan.
3. Bot telegram pada penelitian ini sangat berguna untuk perusahaan pada umumnya dan devisi lapangan pada khususnya ditandai dengan analisis perbandingan sistem lama dengan sistem yang baru.

#### **5.2. SARAN**

Di samping beberapa kesimpulan, penulis juga mencoba memberikan saran yang diharapkan dapat mengembangkan bot Telegram telegram pada penelitian ini, adapun saran dari penulis adalah sebagai berikut:

1. Mengembangkan beberapa rumus agar lebih dapat dimaksimalkan sesuai dengan kebutuhan.
2. Menambahkan fitur hak akses untuk dapat membatasi user guna untuk keamanan data.
3. Menambahkan beberapa fitur masukan data jika ada data yang tidak valid.