

BAB III

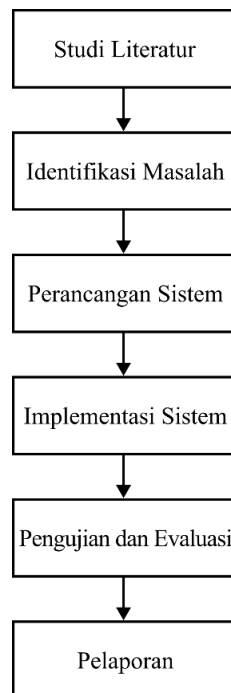
METODOLOGI PENELITIAN

Penelitian ini memiliki Metodologi yang menjelaskan mengenai tahapan penelitian dari mulai bagaimana penelitian akan dilakukan, berbagai kebutuhan perangkat dan rancangan sistem yang akan diimplementasikan dengan menjelaskan proses perancangan pada bot chat di telegram dengan menggunakan metode *web scraping*.

Hasil yang ditargetkan dari penelitian ini adalah penerapan program yang dibuat untuk kepada aplikasi telegram bot chat yang mampu melakukan tindakan secara otomatis untuk mengirimkan request pengguna berupa dokumen sesuai dengan data DOI/url artikel yang diinginkan oleh *user*.

3.1 Prosedur Penelitian

Penelitian ini membutuhkan prosedur penelitian agar pembuatan sistem yang dirancang dapat terukur dari segi proses dan juga hasil. Dilakukan dan dilaksanakan berdasarkan pendekatan baku dengan cara yang telah diatur seperti teknik dalam pengumpulan data dan juga pertanyaan-pertanyaan yang disusun untuk proses tanya jawab dengan narasumber yang memiliki keilmuan dan pengalaman yang berkaitan dengan penelitian yang sedang dilakukan. Tahapan kerja dalam penelitian ini dijelaskan dalam gambar 3.1.



Gambar 3.1 Tahapan Penelitian

Pada tahap pertama penelitian (Gambar 3.1) yaitu tahap studi literatur dilakukan dengan tujuan untuk mendapatkan referensi yang valid dan dapat dipertanggungjawabkan dengan ruang lingkup penelitian yang merancang bot chat telegram atau metode *web scraping*. Informasi yang didapatkan akan diterapkan dalam perancangan dan pengembangan sistem telegram bot. Teknik *web scraping* diimplementasikan dengan bahasa pemrograman python sesuai dengan studi literatur.

Tahap kedua adalah identifikasi masalah. Tujuan dari identifikasi masalah ialah menentukan problem yang ditemukan dari tahap studi literatur. Pertanyaan-pertanyaan kemudian ditentukan untuk mendapatkan tujuan dari penelitian yang dilakukan. Pertanyaan tersebut menjadi dasar bagaimana penelitian ini dapat mencapai tujuannya. Perancangan sistem, menentukan alur dan bagaimana sistem

dapat bekerja dilakukan untuk menjawab masalah dan pertanyaan yang telah dibuat dengan implementasi chat bot Telegram yang mampu mengunduh paper dan artikel ilmiah secara banyak sekaligus. Sistem yang telah diterapkan membutuhkan pengujian untuk memastikan program yang dibuat berjalan sesuai dengan rancangan. Pengujian ini melihat respon apakah sesuai atau tidak dengan rancangan juga menguji waktu respon sistem dari bot chat untuk scraping data dari halaman website dengan menghitungnya menggunakan alat ukur waktu seperti *stopwatch*. Hasil dari pengukuran akan dihitung waktu rata-rata respon dengan membedakan respon yang hanya memuat teks dan respon yang memuat file paper dan artikel ilmiah.

3.2 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan merupakan tahapan lanjutan setelah menemukan solusi dari permasalahan yang perlu dilakukan untuk melakukan sebuah penelitian. Analisis kebutuhan akan menjadi dasar dari sebuah penelitian yang akan dilakukan.

Analisis kebutuhan yang dilakukan juga meliputi uraian kebutuhan seperti apa yang akan digunakan sebagai alat penelitian, kebutuhan apa saja yang akan digunakan dalam melakukan pengujian sehingga permasalahan-permasalahan yang akan dihadapi sudah disiapkan guna sistem yang telah diterapkan dapat berfungsi sesuai dengan fitur yang telah diprogram. Terdapat 2 bagian dalam analisis kebutuhan non fungsional, yakni analisis kebutuhan *hardware* dan analisis kebutuhan *software*:

3.2.1 Analisis Kebutuhan *Hardware*

1. Komputer Server dengan spesifikasi :
 - a) Processor AMD A Radeon R5
 - b) HDD (*hard disk drive*) 500GB (*giga byte*)
 - c) RAM (*random access memory*) 2GB (*giga byte*)
2. Laptop dengan spesifikasi :
 - a) OS (*operating system*) Windows 7 Ultimate 64-bit
 - b) Processor AMD A Radeon R5
 - c) SSD (*solid state drive*) 250GB (*giga byte*)
 - d) RAM (*random access memory*) 4GB (*giga byte*)
3. Smartphone dengan spesifikasi
 - a) OS (*operating system*) Android 9.0 (*Pie*)
 - b) Chipset Qualcomm SM8150 Snapdragon 855 (7 nm)
 - c) Internal memory 64GB (*giga byte*)
 - d) RAM (*random access memory*) 6GB (*giga byte*)

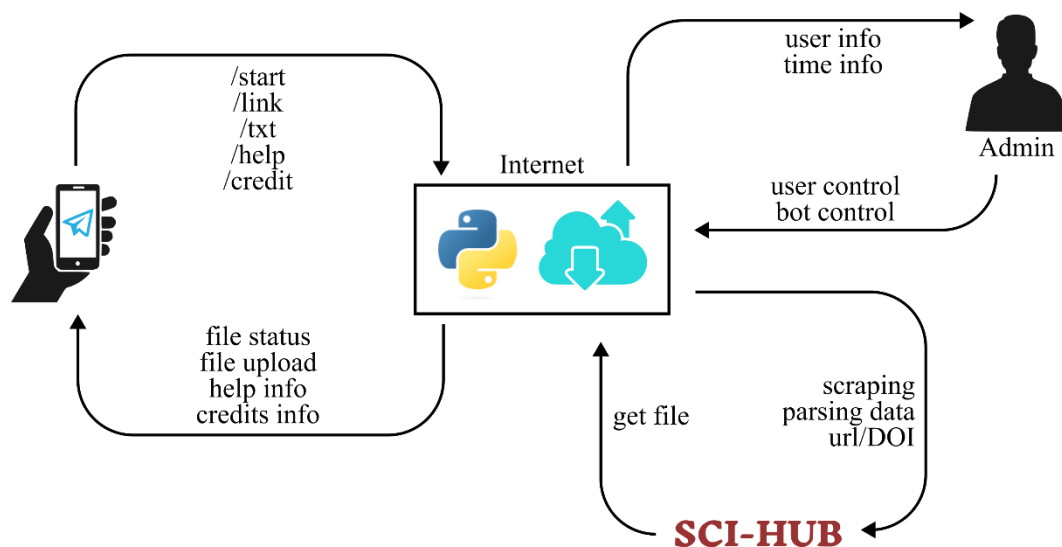
3.2.2 Analisis Kebutuhan *Software*

1. Visual Studio Code
2. Phyton
3. Command Prompt
4. Telegram

3.3 Perancangan Sistem

Pada tahapan perancangan sistem dilakukan penentuan bagaimana sistem memecahkan masalah yang diteliti pada objek penelitian ini. Desain sistem

memberikan gambaran tentang apa yang harus dilakukan dan bagaimana sistem bot Telegram bekerja di Telegram Messenger menggunakan metode *web scraping*. Merepresentasikan garis besar rancangan sistem yang akan dibuat pada penelitian ini. Desain perancangan sistem yang akan dibangun pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Perancangan Sistem

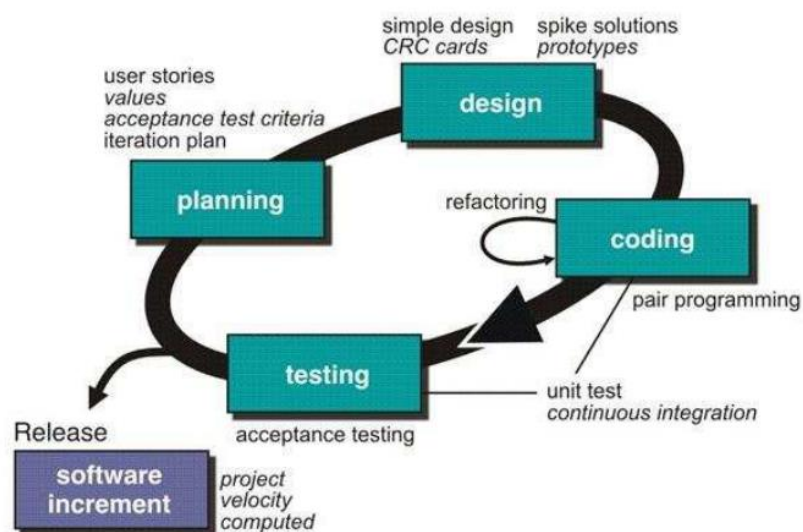
Gambar 3.2 menunjukkan terdapat proses yang saling keterkaitan antara user, admin, dan website yang terhubung ke server. User membuka bot chat kemudian mengirim perintah yang diteruskan ke server dan terhubung dengan web *sci-hub* mengambil data dengan metode *web scraping* (*CSS selector*) yang dimuat oleh sistem.

Pengguna melakukan permintaan melalui menu atau input manual di Telegram bot dengan username @andrillh_bot yang dibuat dengan menggunakan fitur dari BotFather. Perintah tersebut merupakan fitur Telegram Messenger kepada

pengembang untuk membuat bot Telegram. Server menerima pesan dan meneruskannya ke bot server. Robot server yang telah di running dengan program yang telah dibuat memproses pesan untuk dapat menangani perintah dengan jawaban yang benar kepada pengguna dalam bentuk teks atau dokumen. Respon yang dikirim oleh bot sudah ditangani oleh program yang disimpan di server, dengan begitu bot akan menjawab sesuai dengan perintah/*command* user.

3.4 Metode Pengembangan Sistem

Dalam pengembangan Aplikasi Bot Chat Telegram SAD (*Scientific Article Downloader*), model yang digunakan adalah Metode Extreme Programming (Patel, 2019).



Gambar 3.3 Kerangka Kerja Extreme Programming

3.4.1. Planning

Fokus dari perancangan ini untuk mendapatkan gambaran tentang fitur dan fungsionalitas perangkat lunak yang akan dibangun. Perancangan dimulai dengan

pembuatan kumpulan gambar atau alur berupa bagaimana pengguna memakai aplikasi, yang berfungsi sebagai deskripsi dasar perangkat lunak. Kumpulan gambar atau cerita disusun menjadi sebuah indeks, setiap item memiliki prioritasnya masing-masing. Memperkirakan waktu dan biaya sebagai cost dari pengerjaan perangkat lunak. Pengembangan aplikasi dimulai dengan menyusun tugas pengembangan yang berupa timeline dan progress. Selama proses pengembangan perangkat lunak, peneliti dapat mengubah desain apapun dari aplikasi yang dibuat. Perubahan dapat dilakukan dengan cara mempertimbangkan semua kemungkinan agar perangkat lunak yang dirancang dapat lebih optimal sesuai dengan kebutuhan yang telah digambarkan.

3.4.2. Design

Design bertujuan untuk memberikan gambaran alur logika yang akan ditempelkan di dalam sistem. Aplikasi yang memiliki desain dapat membuat keterkaitan antara proses berkurang. Fungsi pada sistem dengan begitu akan lebih stabil dan tidak terlalu bergantung satu sama lain yang mengakibatkan sistem tidak akan mengalami error secara keseluruhan. Aktivitas desain dibutuhkan untuk menjadi panduan saat membangun dan mengembangkan software sesuai dengan paparan pada rumusan masalah yang kemudian disusun ke dalam rancangan untuk digunakan di tahap *planning*. Proses ini berlangsung secara terus menerus dalam siklus XP.

3.4.3. Coding

XP merekomendasikan agar tim terlebih dahulu membuat modul pengujian unit yang bertujuan untuk menguji setiap fitur dan deskripsi yang dibutuhkan oleh

pengguna. Proses coding dilakukan setelah berbagai uji coba dan penyesuaian pada desain dan juga fitur yang diinginkan. Coding dilakukan dengan menggunakan aplikasi Visual Studio Code dengan bahasa pemrograman python yang menerapkan metode *web scraping* dengan jenis css selector agar dapat mengambil file multimedia yang telah di 'embed' di dalam website *sci-hub*. Program memuat library os yang dibutuhkan untuk melakukan akses terhadap direktori server. Library TeleBot untuk mengakses API Telegram Bot. Library request untuk membuat permintaan HTTP ke situs web *sci-hub* yaitu berupa halaman artikel atau paper. Library BeautifulSoup untuk melakukan parsing HTML dan melakukan ekstraksi berupa file multimedia yaitu dalam bentuk PDF yang kemudian akan diteruskan ke pelanggan. Proses penulisan kode berlangsung selama 14 hari dengan melakukan uji coba secara berkala untuk memastikan kode yang dibuat dapat berjalan.

3.4.4. Testing

Tahapan uji coba ini bertujuan untuk memastikan bahwa program yang telah dibuat tidak memiliki kecacatan (bug) dan meminimalisir error yang terjadi pada program agar dapat berjalan dengan fitur yang telah dirancang. Proses yang diuji adalah sisi user dan admin. Fitur yang dimiliki oleh bot chat @andrillh_bot akan diuji apakah merespon sesuai desain yang dirancang atau tidak. Setiap perintah di dalam bot memiliki respon berbeda-beda. Admin dibuatkannya bot tersendiri yaitu untuk memberikan informasi berupa user baru yang mengakses bot. Di dalam testing juga diuji seberapa cepat bot merespon terhadap perintah yang diberikan oleh user.

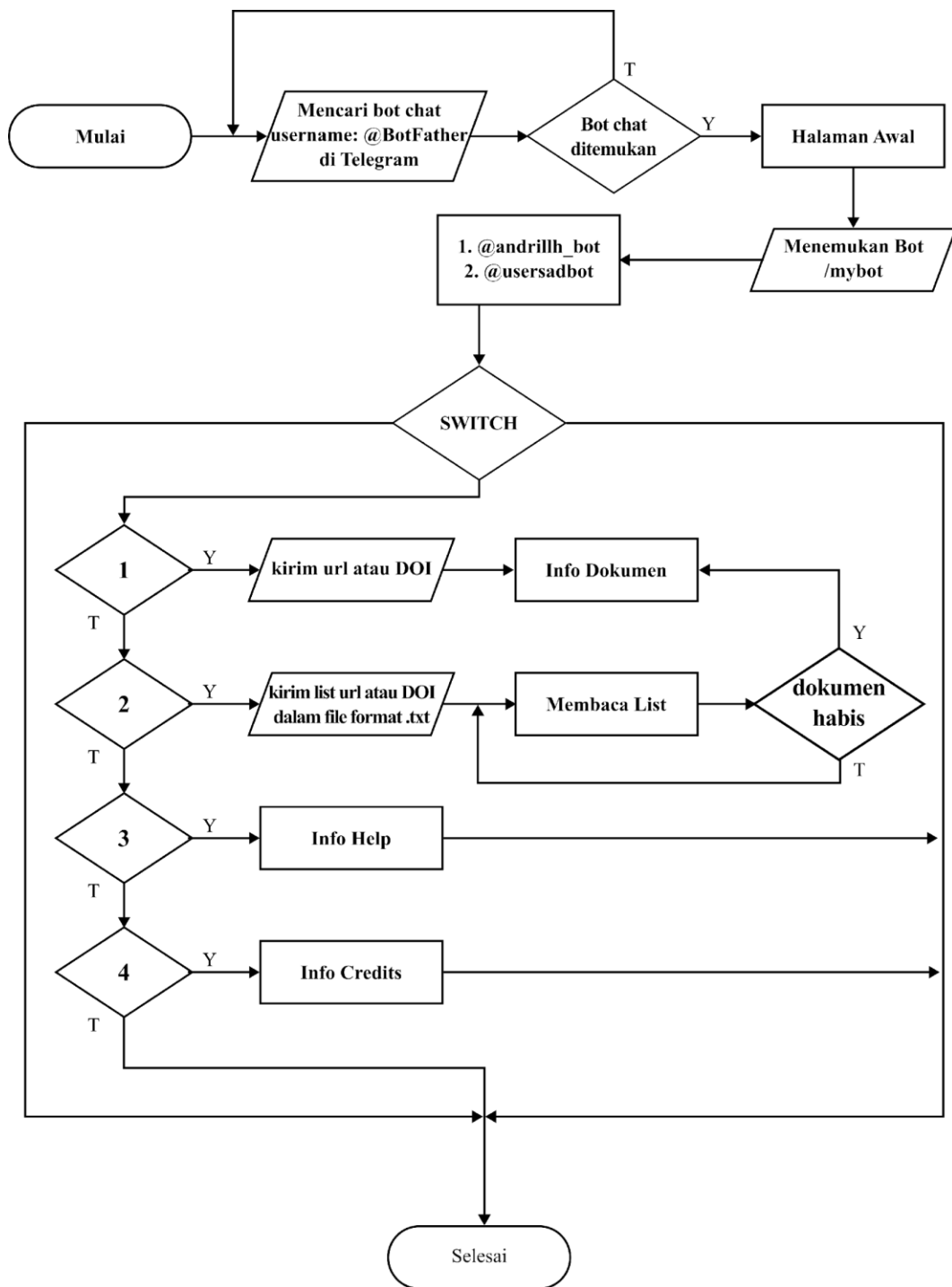
3.5 Flowchart Diagram

Flowchart diagram dibuat untuk memberikan penjelasan secara singkat bagaimana suatu proses atau alur dalam penggunaan bot chat dapat dilakukan. Alur yang dijelaskan bertujuan untuk membantu user agar dapat memahami cara kerja bot, dan juga fitur apa saja yang dapat dilakukan oleh bot. Proses pendaftaran telegram menjadi tidak kalah penting dalam penelitian ini sebagai bentuk antisipasi bagi pengguna yang tidak memiliki aplikasi telegram messenger.

Flowchart yang dibuat tidak hanya dari sisi user, tetapi sisi admin juga terdapat proses. Dalam aplikasi ini admin memiliki proses untuk menerima notifikasi dari bot mengenai user baru. Flowchart sistem Unsil Scientific Article (USA) Downloader untuk admin juga terdapat proses untuk mengatur bot chat dari mulai nama, keterangan, hingga respon bot terhadap user yang terdapat di dalam server.

3.5.1 Flowchart Penggunaan Bot

Pembuatan akun telegram telah selesai kemudian pengguna dapat mulai untuk menggunakan bot Telegram yang dirancang khusus untuk mengunduh artikel atau *paper* dari berbagai sumber jurnal. Pertama, pengguna mencari ID bot chat bernama **@andrillh_bot** atau dengan nama *Scientific Article Downloader* melalui aplikasi Telegram. Atau dengan link *t.me/andrillh_bot* Proses penggunaan bot dapat dilihat dalam Gambar 3.4.



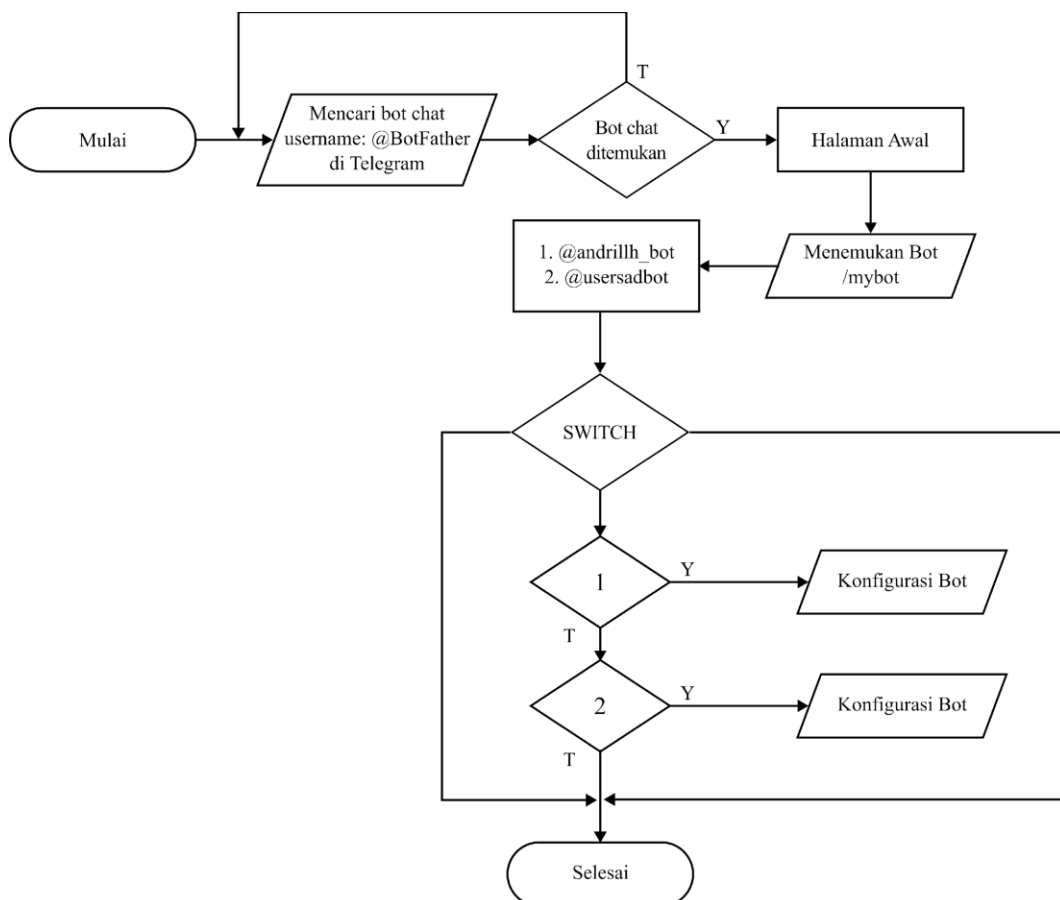
Gambar 3.4 Flowchart Menggunakan Bot

Dalam penggunaan bot telegram (Gambar 3.4) ketika bot @andrillh_bot sudah ditemukan, maka user akan diarahkan kepada tampilan awal berupa ruang chat dengan bot yang berisi sedikit info bagaimana cara menggunakan bot yaitu

perintah untuk memulai percakapan dengan perintah /start atau menekan tombol start bagi pengguna *handphone*. Setelah memulai percakapan bot akan menyapa user dengan menyebut nama, kemudian mengarahkan untuk mulai memakai bot chat dengan perintah yang tersedia, yaitu unduh satu artikel atau *paper*, unduh banyak artikel dan/atau *paper*, info cara menggunakan bot dan credit. Setiap perintah akan direspon sesuai dengan perintah yang diinput oleh *user*.

3.5.2 Flowchart Konfigurasi Bot

Selain pengguna, bot ini juga dapat dioperasikan oleh admin untuk mengatur bagaimana bot dapat merespon setiap pesan yang dikirimkan oleh pengguna.



Gambar 3.5 Flowchart Konfigurasi Bot

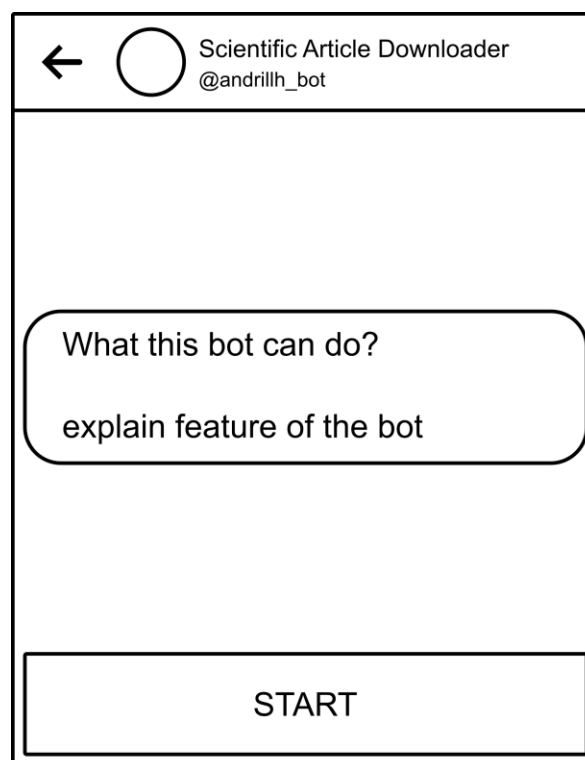
Sistem bot yang dibuat untuk admin adalah untuk mengontrol input user, mengontrol info bot, dan juga mendapatkan info user baru dan juga waktu kapan user mengakses bot.

3.6 Desain Interface

Desain interface yang dibuat mengikuti tampilan di dalam aplikasi Telegram untuk versi mobile, desktop maupun websitenya.

3.6.1. Tampilan Awal Bot

Tampilan awal dari desain interface telegram secara default memiliki desain seperti Gambar 3.6.



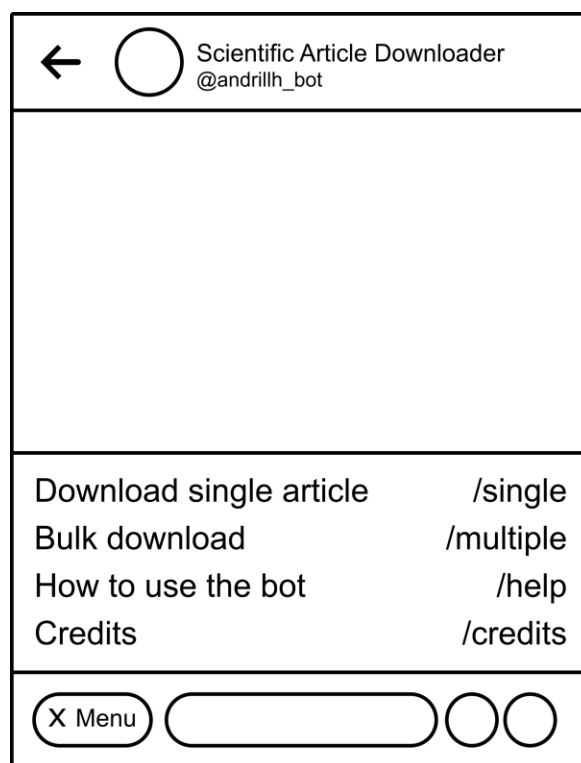
Gambar 3.6 *Interface* awal bot Telegram

Dalam desain ini memiliki tampilan berupa nama bot di bagian atas yakni bot dinamai sebagai *SAD* yang merupakan kepanjangan dari *Unsil Scientific Article*

Downloader. Di Bagian tengah merupakan penjelasan singkat mengenai kegunaan bot. Tombol start di bagian bawah merupakan salah satu perintah user untuk memulai percakapan dengan bot.

3.6.2. Tampilan Menu Bot

Memulai percakapan dengan bot akan membuka pesan box untuk user dapat menginput pesan secara manual yang disertai dengan menu lainnya. Menu-menu ini berupa list command bot di bagian kiri *text box* dan di bagian kanan terdapat menu input suara dan file seperti gambar 3.8:



Gambar 3.7 Tampilan Menu Bot

Saat menekan menu perintah akan menampilkan list berupa menu unduh satu artikel, unduh banyak, cara menggunakan bot, dan credit. Menu unduh satu artikel atau *paper* adalah perintah yang kemudian bot akan meminta user

memasukan url atau DOI dari artikel yang ingin diunduh. Menu unduh banyak merupakan perintah yang akan di respon bot dengan meminta list url atau DOI dari banyak artikel yang ingin di unduh user dengan format file **.txt*. Menu cara menggunakan bot adalah instruksi secara singkat bagaimana user menggunakan bot. Menu terakhir adalah credits yang akan menampilkan identitas orang-orang yang berjasa dalam pembuatan aplikasi bot ini.

3.7 Implementasi Sistem

Pada fase ini, desain sistem diimplementasikan dalam python dengan menggunakan Visual Studio. Sistem yang dibuat adalah bot Telegram yang menggunakan metode *web scraping* dengan menggunakan library pada pemrograman Python. Proses implementasi dimulai dengan mengembangkan desain sistem, yaitu. flowchart, dan penulisan program yang akan diimplementasikan dalam sistem. Antarmuka pengguna kemudian diterapkan pada desain tampilan dengan pengguna.

3.8 Metode Pengujian

Pada fase ini dilakukan beberapa tes uji yang pertama menguji fungsi fitur yang tersedia dan respon dari input user telah sesuai. Pengujian kedua dilakukan dengan mengukur waktu pada saat proses *request* saat pengguna mengirim perintah pada telegram bot, yang melalui server. Uji coba merupakan seberapa lama waktu saat proses *request* hingga bot merespon dengan mengirimkan balasan, dengan menggunakan *stopwatch* dan dilakukan sebanyak 100 kali untuk setiap respon. Hasil pengujian yang didapat adalah kecepatan atau respon dari telegram bot.