

INTEGRASI SISTEM INFORMASI AKADEMIK MAHASISWA DENGAN BOT TELEGRAM SEBAGAI MESIN PENJAWAB OTOMATIS

INTEGRATION OF STUDENT ACADEMIC INFORMATION SYSTEMS WITH TELEGRAM BOT AS AN AUTOMATIC ANSWERING MACHINE

Harminto Mulyo^{1*}, Gunawan Mohammad²

^{1,2}Universitas Islam Nahdlatul Ulama Jepara
Email : ^{1*}minto@unisnu.ac.id, ²gunawan@unisnu.ac.id
*Penulis Korespondensi

Abstrak - Sistem Informasi Akademik Mahasiswa (SIAMA) adalah perangkat lunak berbasis web yang dikembangkan untuk mengelola data akademik mahasiswa. Di sistem ini, data akademik mahasiswa diolah untuk menjadikan informasi akademik yang bermanfaat bagi mahasiswa. Namun, untuk mengakses sistem tersebut membutuhkan banyak langkah sehingga waktu kurang efisien jika harus menelusuri seluruh tahapan tersebut. Belum lagi kebutuhan kuota data internet yang dihitung berdasarkan seberapa besar konten system tersebut yang di unduh dan unggah. Untuk mempercepat akses terhadap informasi tersebut, dapat dilakukan dengan mengintegrasikan layanan Bot Telegram sebagai layanan penjawab otomatis yang dikendalikan oleh *robot*. Dengan menggunakan metode PPDIOO, SIAMA telah berhasil diintegrasikan sebagai mesin penjawab otomatis melalui pemanfaatan dan pengembangan *Library PHP Telegram Bot* dikolaborasikan dengan *Crontab Linux*. Dari hasil pengujian, diketahui waktu respon dari Bot ditunjukkan pada jam dan menit yang sama atau < 1 menit yang artinya waktu eksekusi Bot relative sangat cepat.

Kata kunci: *bot, bot telegram, sistem informasi akademik mahasiswa, long-polling method*

Abstract - *The Student Academic Information System (SIAMA) is web-based software developed to manage students' academic data. In this system, student academic data is processed to make academic information useful for students. However, to access the system requires many steps so that time is less efficient if you have to browse through all the stages. Not to mention the need for internet data quotas that are calculated based on how much content the system downloads and uploads. To speed up access to that information, it can be done by integrating the Telegram Bot service as an automated answering service controlled by robots. Using the PPDIOO method, SIAMA has been successfully integrated as an automated answering machine through the utilization and development of the Telegram Bot PHP Library collaborated with Crontab Linux. From the test results, it is known that the response time from the Bot is indicated at the same hour and minute or < 1 minute which means that the bot execution time is relatively fast.*

Keywords: *bot, telegram bot, student academic information system, metode long-polling*

1. PENDAHULUAN

Dukungan teknologi terhadap pemanfaatan/penerapan sistem informasi sangat penting bagi keberlangsungan suatu universitas. Sistem informasi di suatu universitas dibangun guna memenuhi layanan akademik terkomputerisasi untuk menunjang dan meningkatkan kecepatan, kualitas layanan dan daya saing institusi.

Informasi akademik berbasis website atau portal dapat diakses oleh mahasiswa menggunakan *web browser*. Namun, cara mengakses informasi seperti ini kurang efektif jika mahasiswa hanya menghendaki informasi yang spesifik misalnya, hanya ingin melihat Kartu Hasil Studi (KHS). Akses ke sistem informasi akademik, biasanya dilengkapi dengan hak akses tertentu yang dapat memfilter siapa saja yang diperbolehkan masuk ke dalam sistem, dan fitur apa saja yang ditetapkan sistem terhadap pengguna tersebut sehingga untuk mendapatkan informasi KHS tersebut, mahasiswa harus berhasil masuk dan menerima/download data dari server beserta tampilan yang kadang tidak dibutuhkan.

Layanan informasi nilai mahasiswa telah berhasil diterapkan dengan memanfaatkan teknologi SMS Gateway dan *Trigger Database* [1]. Pada penelitian tersebut, permintaan informasi dari mahasiswa akan dikirim dan diolah di *Gammu*, selanjutnya *gammu* akan melakukan *request* ke *database* untuk menentukan jawaban dari *request* yang

dikirim tersebut. Jawaban dari *trigger* akan dikirim kembali ke *Gammu* untuk diteruskan ke pengguna. Pada penggunaan teknologi ini, akan sangat tergantung pada operator *selluler* yang digunakan mahasiswa, karena bisa jadi masing-masing mahasiswa menggunakan jasa operator yang berbeda-beda sehingga biaya pulsa untuk penerimaan *request* dan pengiriman balik akan tergantung pada tarif yang ditetapkan oleh operator tersebut.

Pada penelitian lain, berhasil mengembangkan *bot telegram* sebagai media informasi untuk mengakses data penelitian [2]. Pada penelitian ini, permintaan data penelitian akan dijawab otomatis oleh bot telegram dengan mengirimkan data atau arsip penelitian dalam bentuk file.

Sistem informasi akademik (SIKAD) telah berhasil diterapkan di banyak universitas. SIKAD diterapkan untuk mempercepat proses pengolahan data dan pelaporan ke kemdikbud. Disisi lain, data yang telah diolah tersebut harus dimanfaatkan untuk kepentingan internal, tidak terkecuali di Universitas Islam Nahdlatul Ulama' Jepara (UNISNU Jepara). Di UNISNU Jepara, telah berhasil menerapkan Sistem Informasi Akademik Mahasiswa (SIAMA) sebagai bagian dari SIKAD terpadu mulai tahun 2013 [3], dan terus dikembangkan seiring dengan kemajuan teknologi dan kebutuhan yang ada. Di SIKAD pula telah berhasil diintegrasikan layanan *auto respons* dengan memanfaatkan teknologi *SMS Gateway* sebagai jalur pengiriman informasi secara cepat.

Layanan *SMS Gateway* tergantung pada syarat dan ketentuan layanan dari operator penyedia jasa layanan. Beberapa operator layanan *selluler* tidak memperbolehkan pengiriman sms dengan jumlah banyak di waktu yang sama, karena hal ini dianggap sebagai tindakan *spamming*.

Untuk mengatasi permasalahan tersebut di atas, pada penelitian ini bermaksud untuk mengintegrasikan Sistem Informasi Akademik yang sudah ada dengan informasi yang telah diolah pada sistem tersebut supaya dapat diakses melalui layanan Bot Telegram untuk mempercepat *reply* terhadap informasi akademik mahasiswa.

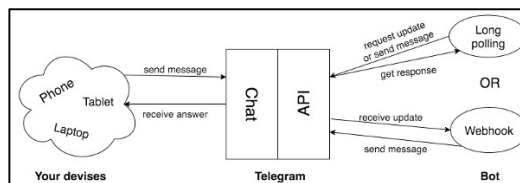
Sistem Informasi Akademik (SIA) merupakan suatu perangkat lunak yang berguna untuk manajemen administrasi pendidikan maupun sistem administrasi mahasiswa seperti penilaian proses pembelajaran, proses pendaftaran kegiatan, penjadwalan kegiatan pembelajaran, administrasi tagihan pembayaran, pengambilan kartu rencana studi, informasi hasil studi, kontrol kehadiran atau presensi, dan sebagainya [4].

Perkembangan teknologi informasi dan komunikasi membawa dampak terhadap perangkat lunak sebagai media komunikasi. Perubahan fasilitas telepon genggam sebagai media komunikasi yang semula hanya digunakan untuk mengirim/menerima pesan melalui sms dan melakukan/menerima panggilan sekarang berkembang menjadi aplikasi pesan instan/*instant messaging* (IM) dengan fitur tambahan yaitu *video call*. Diantara aplikasi IM yang berkembang saat ini yaitu Whatsapp, Line, Snapchat, Facebook Messenger dan Telegram [5]. Sampai saat ini, aplikasi Telegram adalah satu-satunya aplikasi yang menyediakan *Application Programming Interface* (API) secara terbuka untuk bisa dimanfaatkan oleh pengembang aplikasi dalam menerapkan robot atau dimanfaatkan untuk system informasi [6]. API atau *Application Programming Interface* adalah sebuah *interface* yang berperan menjadi penghubung antara satu aplikasi dengan aplikasi lainnya baik dalam *platform* yang sama atau *cross-platform* [7].

Bot Application Programming Interface (API) adalah teknologi *open source* yang disediakan oleh Telegram untuk membangun aplikasi bot Telegram bagi developer. Bot API ini merupakan antarmuka berbasis HTTP untuk menghubungkan bot yang dikembangkan oleh *developer* dengan sistem Telegram (Telegram Bot API, <http://core.telegram.org/bots/api>).

Bot Telegram adalah *robot* dalam layanan pesan aplikasi Telegram yang dapat dirancang sedemikian rupa sehingga dapat berfungsi untuk membantu mengotomatiskan tugas-tugas manusia. Bot Telegram adalah akun khusus yang tidak memerlukan nomor telepon. Akun ini berfungsi sebagai antarmuka untuk menjalankan kode yang telah dibangun. Untuk keamanan data, server perantara di Telegram akan menangani semua enkripsi dan komunikasi dengan Bot API. Sehingga developer tidak perlu mengetahui cara kerja protokol enkripsi *MTPProto* di server Telegram.

Ada dua cara untuk membuat Bot Telegram, menggunakan metode *Long-Polling* dan metode *Webhook*. Dengan metode *Long-Polling*, *server* memeriksa bot secara berkala jika ada pesan masuk. Jika ada pesan yang masuk, *server* akan mengeksekusi berdasarkan pesan request yang dikirimkan oleh pengguna. Jika tidak ada pesan, kondisi *server idle*. Dalam metode *Long-Polling* Bot akan berjalan langsung melalui server, sementara dalam metode *Webhook* membutuhkan layanan *hosting https* [8].



Gambar 1. Arsitektur Bot Telegram

Agar Bot dapat berjalan dengan baik, minimal dibutuhkan 2 komponen untuk dapat bekerja dan memenuhi tujuannya. Komponen pertama adalah *script* PHP yang menjalankan bot tersebut. Antara lain apa yang akan dilakukan bot jika menerima perintah tertentu dari pengguna. Komponen kedua adalah *script* PHP untuk dijalankan secara *continue* [9].

Untuk dapat memfungsikan Bot Telegram, bot tersebut harus dibuat terlebih dahulu di *BotFather*. *BotFather* adalah bot telegram yang difungsikan untuk membuat bot baru dan mengubah pengaturan untuk yang sudah ada atau sudah pernah didaftarkan sebelumnya.

Untuk menjalankan perintah secara periodik diperlukan sebuah *crontab* (*Cron Table*) *file*. *Crontab* ini merupakan *file* konfigurasi yang berisi perintah yang akan dijalankan secara periodik berdasarkan pengaturan jadwal yang telah ditetapkan [10]. Kontrol untuk mengatur waktu akan diatur dengan menggunakan PHP (*Hypertext Preprocessors*) sebagai bahasa pemrograman, di mana PHP adalah bahasa yang peneliti gunakan untuk membuat server menghasilkan *output* yang berbeda setiap kali browser melakukan *request* terhadap suatu halaman [11]. File php tersebut berisi kode program untuk membaca basis data dan bahkan untuk memanggil perintah trigger yang tertanam di basis data. Trigger adalah Pemicu adalah tugas yang dijalankan sebagai tanggapan terhadap beberapa peristiwa database yang telah ditentukan, seperti setelah baris baru ditambahkan ke tabel tertentu. Secara khusus, peristiwa ini melibatkan penyedpan, modifikasi, atau penghapusan data tabel, dan tugas dapat terjadi baik sebelum atau segera setelah peristiwa tersebut [12].

Merujuk pada website resmi telegram, ada beberapa contoh kode bot atau *library bot code* dengan berbagai Bahasa pemrograman [13]. Pustaka tersebut mendukung metode *webhook* maupun *long-polling* sehingga proses pengembangan bot server akan lebih cepat dibandingkan mengembangkan mulai dari awal.

2. METODE PENELITIAN

Dalam penelitian ini, peneliti menerapkan metode PPDIIO (*Prepare, Plan, Design, Implementation, Operate, Optimize*). Metode ini dirancang dan diterapkan oleh Cisco untuk mendukung jaringan berkembang [14]. Metode ini dipilih karena tahapan-tahapan yang ada pada metode ini mengandung unsur yang tepat untuk diterapkan terutama pada model perancangan sistem komunikasi, yaitu komunikasi data dari basis data SIAKAD ke bot telegram.

2.1 Prepare

Pada proses persiapan, bot baru perlu diciptakan/didaftarkan ke Telegram dengan memanfaatkan layanan *BotFather*. Perintah untuk membuat bot baru yaitu dengan mengetikkan *command /newbot* pada *BotFather*. Selanjutnya, *BotFather* akan meminta memasukkan nama Bot dan nama pengguna, kemudian menghasilkan *token* otorisasi untuk bot baru tersebut. Untuk memberikan deskripsi bot digunakan perintah */setdescription* dan bot sudah siap untuk ditetapkan perintah.

2.2 Plan

Pada tahap perencanaan, perlu ditetapkan *command/perintah* apa saja yang akan direspon oleh bot. *Command* tersebut yang nantinya akan dijadikan kata kunci pencarian informasi ke *database*. Untuk menetapkan kata kunci digunakan perintah */setcommands*. Berikut daftar kata kunci yang direncanakan untuk menjawab/merespon *request* dari *user*.

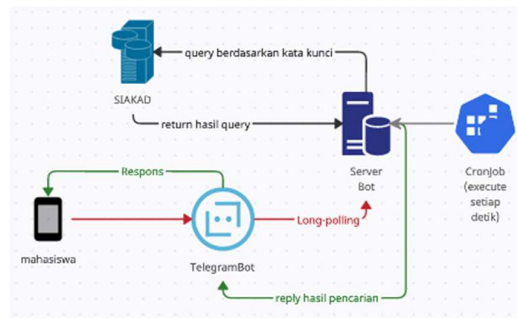
Tabel 1. Daftar keyword Bot

key	keterangan
/registrasi	Registrasi akun SIAMa
/status	Status informasi mahasiswa
/aspirasi	Penyampaian aspirasi mahasiswa
/jadkul	Jadwal kuliah mahasiswa
/persen	Presentase kehadiran perkuliahan
/jaduts	Jadwal UTS
/jaduas	Jadwal UAS
/nilai	Daftar nilai mahasiswa
/pa	Pembimbing akademik mahasiswa
/ps	Pembimbing skripsi mahasiswa

Selain kata kunci atau *command* pencarian ke database, perlu juga dirancang struktur data atau basis data yang akan dibaca atau *query* untuk menghasilkan informasi yang dimaksud. Untuk pembacaan informasi yang lebih cepat, pada sisi database diterapkan beberapa *trigger* sesuai dengan kata kunci yang telah ditetapkan. *Trigger* tersebut ditanam di sisi basis data sehingga dari sisi keamanan bukan pada sisi pemrogramannya, melainkan dari pengamanan basis data.

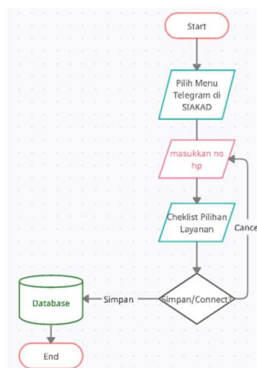
2.3 Design

Pada tahap *design*, dimulai dari *design* Arsitektur. Design ini menggambarkan perangkat teknologi yang akan digunakan, jalur komunikasi/saluran data dan mekanisme penjadwalan eksekusi yang dikendalikan oleh *Crontab/Cronjob*.



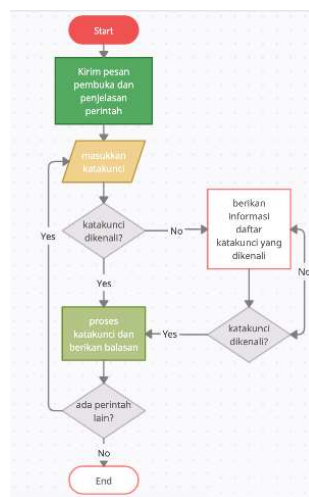
Gambar 2. Arsitektur Bot Telegram

Sebelum pengguna mendapatkan akses ke bot telegram, pengguna diwajibkan untuk mendaftar ke SIAKAD masing-masing. Data ini nanti yang akan digunakan sebagai kunci pencarian informasi kemahasiswaan berdasarkan Nomor Induk Mahasiswa (NIM).



Gambar 3. Alur Registrasi Akun Mahasiswa

Alur komunikasi antara bot server dengan sistem akademik digambarkan dengan diagram alir, mulai dari kata kunci dikirim dari user/pengguna sampai dengan *reply* yang diterima oleh pengguna.



Gambar 4. Alur Send Reply

2.4 Implementation

Bot Telegram diimplementasikan dalam lingkungan bahasa pemrograman PHP ditempatkan di server SIAKAD itu sendiri menggunakan metode *Long-Polling*. Metode *long-polling* punya kendala dalam hal interval waktu yang dibutuhkan untuk membaca/menerima *request*, namun disini server akan lebih stabil karena tidak terbebani *request* yang menumpuk. Untuk mengatasi kekurangan tersebut, interval tersebut bisa diatur menggunakan *contab*. *Crontab/Cron Job* tersebut akan mengeksekusi Modul Telegram untuk dijalankan setiap detik.

2.5 Operate

Tahapan pengoperasian dilakukan dengan menggunakan seperangkat smartphone yang telah di install aplikasi Telegram. Untuk akun telegram telah terdaftar dan sudah difungsikan sebagai *Messenger* atau aplikasi perpesanan yang termasuk ke dalam aplikasi sosial media.

Aplikasi telegram tersebut harus didaftarkan pada menggunakan akun SIAMA untuk mendapatkan kode akses sekaligus untuk merekam data unik beserta id setiap user yang terhubung dengan telegram bot. Id tersebut digunakan untuk mapping data ke *database* dan ke telegram.

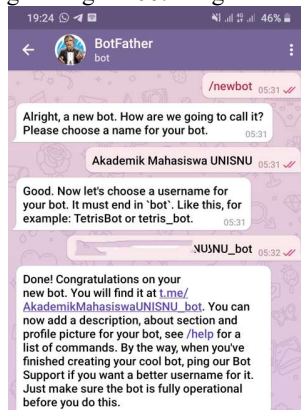
2.6 Optimization

Optimisasi diterapkan pada proses *query database*. Pembacaan data dilakukan dengan Teknik fungsi/function tertentu sehingga pemanggilan tidak perlu dilakukan secara berulang-ulang. Untuk data berelasi, digunakan Teknik *sub-query* di dalam *array multidimensi*, sehingga pembacaan data akan lebih cepat karena proses *looping* yang dikerjakan sistem membutuhkan waktu yang relatif singkat.

3. HASIL DAN PEMBAHASAN

3.1. Pembuatan Bot di BotFather

Bot dibuat di *BotFather* dengan mengetikkan perintah */newbot*. Pada proses ini, *BotFather* meminta untuk memasukkan nama bot dan *username*. Jika pendaftaran berhasil, ada informasi pendaftaran selesai ditandai dengan informasi *token* yang digunakan untuk menghubungkan *bot* dengan SIAMa.



Gambar 5. Membuat Bot Baru di BotFather

Setelah Bot berhasil dibuat, langkah selanjutnya yaitu menetapkan *command* yang digunakan user untuk berkomunikasi dengan Bot. Perintah untuk menetapkan *command* yaitu dengan mengetikkan */setcommands*, selanjutnya daftar *command* dimasukkan/didaftarkan sesuai dengan format yang diinformasikan oleh *BotFather*.

Adapun perintah atau *command* yang perlu ditetapkan atau dimasukkan ke dalam bot tersebut sebagai kata kunci ketika mahasiswa atau pengguna melakukan *request* ke bot seperti tampak pada tabel berikut:

Tabel 2. Keyword Bot dalam BotFather, keterangan

Pembacaan Tabel/Informasi	Parameter	key	keterangan
Tabel <i>contact</i>	Nim	/registrasi	Registrasi akun SIAMa Mahasiswa
Tabel kontak, mhs dan mhs_status	Nim, ta	/status	Informasi Status Mahasiswa
Table kontak, mhs dan aspirasi_mhs	Nim	/aspirasi	Aspirasi mahasiswa
Table kontak, mhs, jadtul	Nim, ta	/jadtul	Jadwal kuliah mahasiswa
Table kontak, mhs, mhs_pre	Nim, ta, kd_makul	/persen	Presentase kehadiran kuliah mahasiswa
Table kontak, mhs, jad_ujian	Nim, ta, jns_ujian	/jaduts	Jadwal Ujian Tengah Semester (UTS)
Table kontak, mhs, jad_ujian	Nim, ta, jns_ujian	/jaduas	Jadwal Ujian Akhir Semester (UAS)
Table kontak, mhs, kuliah, mhs_makul, makul	Nim, ta, makul	/nilai	Daftar nilai mahasiswa
Table kontak, mhs, dosen	Nim	/pa	Pembimbing akademik mahasiswa
Table kontak, mhs, dosen, skripsi	Nim, kd_skripsi	/ps	Pembimbing skripsi mahasiswa



Gambar 6. Menetapkan Command di BotFather

3.2. Implementasi Library PHP untuk Bot Server

Untuk pembuatan Bot Server yang ditempatkan di Server SIAKAD, perlu dipersiapkan database dan *Library PHP* untuk Bot Server. Untuk table *contact* digunakan untuk menyimpan data pengguna yang terdaftar atau bergabung ke bot. Struktur table *contact* seperti gambar 7 berikut:

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default
id_contact	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI	
peer_id	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0
handphone	VARCHAR	16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL
nama_depan	VARCHAR	128	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL
nama_belakang	VARCHAR	128	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL
is_saved	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
try_to_save	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1
last_seen	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL
last_checked	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0000-...
is_aktif	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1

Gambar 7. Struktur Table Contact

Table *contact* di atas akan direlasikan dengan table *login*. Table *login* inilah nanti yang akan menjadi penghubung antara system bot ini dengan SIAKAD. Berikut struktur table *login*.

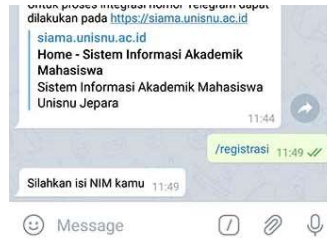
Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default
login	VARCHAR	24	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	MUL	NULL
id_contact	INT	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	MUL	NULL

Gambar 8. Struktur Table Login

Tahap selanjutnya, yaitu generalisasi rancangan alur program ke Bahasa pemrograman PHP memanfaatkan *Library PHP for Bot Telegram* yang dimodifikasi dari <https://github.com/php-telegram-bot/core>. Pada *library* ini, peneliti hanya menambahkan username, *token Bot* serta menambahkan fungsi untuk membaca database SIAKAD. Untuk *Command*, masing-masing perintah yang ada di Bot Telegram didefinisikan masing-masing 1 file *Command* sesuai petunjuk dari *Library* tersebut di atas.

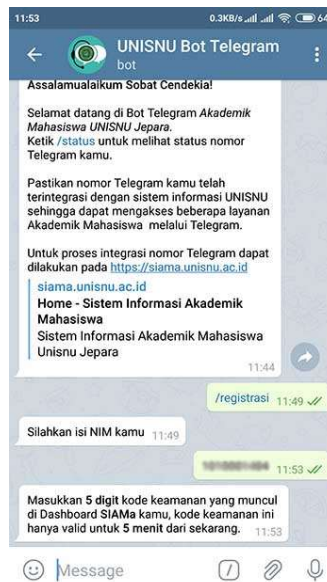
3.3. Pendaftaran Akun Mahasiswa di Bot Telegram

Proses pendaftaran akun Telegram bagi mahasiswa dilakukan pada Bot Telegram, kemudian *Bot Server* akan membuat kode acak atau *token* yang ditampilkan di SIAKAD.



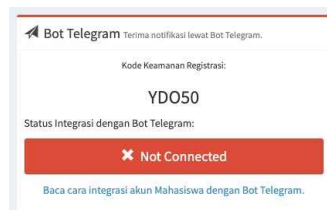
Gambar 7. Proses Registrasi Mahasiswa

Proses selanjutnya, mahasiswa diminta untuk memasukkan NIM sebagai acuan pencarian data mahasiswa ke database mahasiswa.



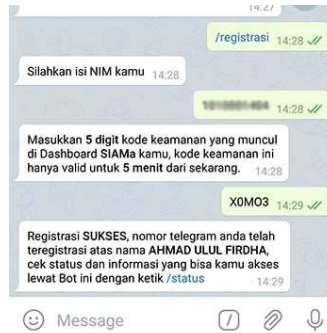
Gambar 8. Proses Pencocokan Token di SIAMA

Setelah NIM diisikan ke Bot, Bot akan meminta 5 Digit kode keamanan yang bisa di lihat di Dashboard SIAMA. Kode kemanan ini dibuat dari pemanggilan fungsi generateRandom() yang nilainya otomatis dimasukkan ke dalam table contact. Nilai ini dijadikan nilai kunci untuk pembacaan 2 arah yaitu dari database dan dari bot telegram.



Gambar 9. 5 Digit Token yang dikirim di SIAMA

5 digit kode keamanan tersebut harus dimasukkan ke bot telegram untuk dilakukan proses registrasi. Digit yang dimasukkan pengguna ke dalam aplikasi Telegram akan dicocokkan dengan contact yang ada di database mahasiswa. Jika kode keamanan ini cocok, selanjutnya *id user* dari telegram akan disimpan ke contact.

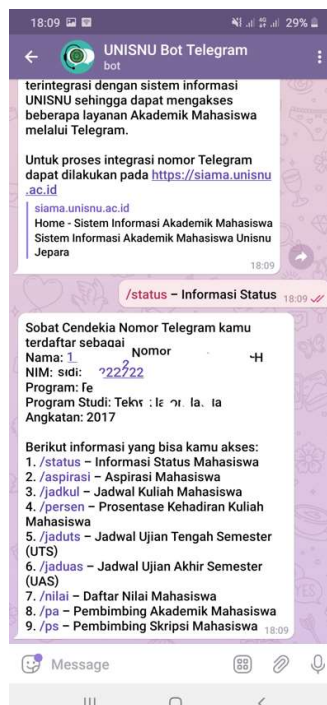


Gambar 10. Informasi NIM berhasil didaftarkan

Jika sudah ada konfirmasi Registrasi SUKSES seperti gambar di atas, maka proses registrasi dan integrasi akun SIAMA dengan Bot Telegram berhasil dilakukan.

3.4. Penggunaan dan Pengujian Bot

Proses penggunaan *Bot Telegram* yaitu dengan mengetikkan *Command* yang sudah tersedia atau dengan memilih daftar *Command* yang muncul di kotak isian Bot.



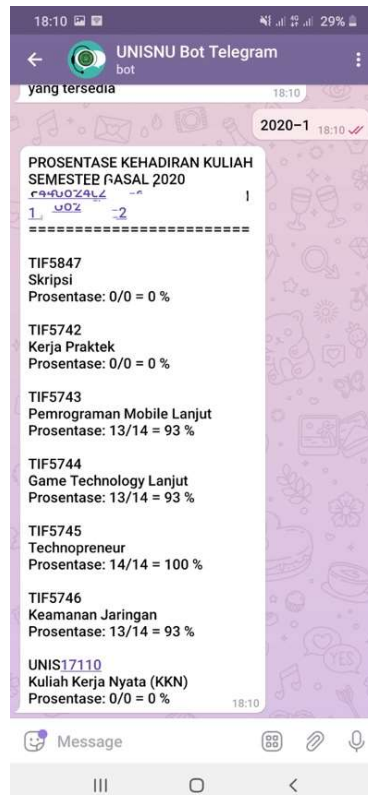
Gambar 11. Percobaan Perintah /status

Pada percobaan di atas, perintah yang dipilih adalah */status*. Perintah tersebut digunakan untuk mengecek status registrasi mahasiswa ke bot. Jika telah terdaftar, maka akan muncul informasi nama, program studi dan lain sebagainya. Waktu pengiriman *request* menunjukkan pukul 18:09 dan *reply* dari *Bot Server* juga pada pukul 18:09 yang artinya *Bot Server* merespon dalam waktu kurang dari 1 menit.



Gambar 12. Percobaan Perintah /persen

Perintah */persen* digunakan untuk mengecek presentase kehadiran mahasiswa dalam perkuliahan. Perintah ini mengandung parameter tambahan yaitu berupa tahun akademik dan sesi (*gasal/genap*) yang tertera di bawah papan ketik perintah. Selanjutnya mahasiswa harus memilih presentase kehadiran di sesi mana yang akan dicari.



Gambar 13. Informasi Persentase Kehadiran

Dari pemanggilan perintah */persen* tersebut di atas, bot menjawab dengan menampilkan seluruh mata kuliah pada semester yang dipilih beserta kehadiran dibagi dengan jumlah pertemuan. Hasil eksekusi program juga menunjukkan jam yang sama yaitu pukul 18:10 yang artinya, request dan respon dari *Bot* terhitung sangat cepat kurang dari 1 menit.

4. KESIMPULAN

Bot Telegram dapat digunakan sebagai antara muka akses ke data SIAKAD dengan memanfaatkan dan mengembangkan *Library PHP Telegram Bot*. Keamanan untuk mengakses informasi data SIAKAD melalui bot dijamin, karena *ChatID* unik untuk setiap pengguna bot. Hasil pengujian antara *request* dan *respon* menunjukkan waktu yang sama sehingga dapat disimpulkan penggunaan waktu untuk *reply* sangat cepat. Penelitian selanjutnya yang dapat dilakukan yaitu bagaimana manajemen *traffic* untuk beberapa *service* di server yang sama sehingga dapat menghindari *traffic overload* pada waktu *peak time*.

UCAPAN TERIMA KASIH

Ucapan terima kasih yang tak terhingga saya haturkan ke mahasiswa Teknik Informatika UNISNU Jepara semester 6 yang telah ikut berpartisipasi dalam pengujian bot telegram ini menggunakan akun atau data pribadi masing-masing sehingga bot ini telah teruji menggunakan data riil mahasiswa yang masih berstatus aktif sebagai mahasiswa.

Tak lupa juga, saya haturkan terima kasih kepada TIM Pusat Data dan Pengembangan IT UNISNU Jepara yang telah membantu penelitian ini dengan ikut menggali informasi yang sering dan sangat penting dibutuhkan mahasiswa dalam waktu yang cepat. Informasi tersebut yang termasuk dijadikan *key* atau kunci pencarian bot Telegram pada penelitian ini.

DAFTAR PUSTAKA

- [1] R. Mayasari, "Sistem Informasi Nilai Mahasiswa Berbasis SMS Gateway menggunakan Trigger pada Database," *Systematics*, vol. 1, no. 1, hal. 44, 2019, doi: 10.35706/sys.v1i1.2010.

- [2] A. D. Mulyanto, "Pemanfaatan Bot Telegram Untuk Media Informasi Penelitian," *Matics*, vol. 12, no. 1, hal. 49, 2020.
- [3] F. Rohman, H. Mulyo, A. Zulfah, dan I. M. Nur, *Sistem Informasi Akademik (SIKad) Universitas Islam Nahdlatul Ulama (UNISNU) Jepara*. Jepara, 2016.
- [4] D. Demirkol, C. Seneler, T. Daim, dan A. Shaygan, "Measuring emotional reactions of university students towards a Student Information System (SIS): A Turkish university case," *Technol. Soc.*, vol. 63, hal. 101412, 2020, doi: <https://doi.org/10.1016/j.techsoc.2020.101412>.
- [5] J. Gregorio, A. Gardel, dan B. Alarcos, "Forensic analysis of Telegram Messenger for Windows Phone," *Digit. Investig.*, vol. 22, hal. 88–106, 2017, doi: <https://doi.org/10.1016/j.diin.2017.07.004>.
- [6] B. N. Prastowo, N. A. S. Putro, dan O. A. Dhewa, "PLO User Interface based on Telegram Bot," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 1, hal. 21, 2019, doi: 10.22146/ijccs.29089.
- [7] B. Cooksey, *An Introduction to APIs*. __: Zapier, Inc., 2014.
- [8] H. Setiaji dan I. V Papatungan, "Design of Telegram Bots for Campus Information Sharing," *IOP Conf. Ser. Mater. Sci. Eng. Pap.*, 2018, doi: 10.1088/1757-899X/325/1/012005.
- [9] A. Cokrojoyo, J. Andjarwirawan, dan A. Noertjahyana, "Pembuatan Bot Telegram Untuk Mengambil Informasi dan Jadwal Film Menggunakan PHP," *J. Infra*, vol. 5, no. 1, hal. 224–227, Program Studi Teknik Informatika Fakultas, 2017, [Daring]. Tersedia pada: <http://studentjournal.petra.ac.id/index.php/teknik-informatika/article/view/5163>.
- [10] A. Wirawan, "Sistem Scheduling Pelaporan Data Akademik di UIN Sunan Kalijaga ke Pangkalan Data Pendidikan Tinggi (PDDikti) dengan Menggunakan Fitur Cron Job di Linux," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 5, no. 3, hal. 177, 2020, doi: 10.14421/jiska.2020.53-05.
- [11] R. Nixon, *Learning PHP, MySQL, JavaScript, CSS & HTML5, 3rd Edition*. O'Reilly Media, Inc, 2014.
- [12] F. M. Kromann, *Beginning PHP and MySQL*. Berkeley: Apress, Berkeley, CA, 2018.
- [13] P. Durov dan Nikolai, "Bot Code Examples," *Telegram*. <https://core.telegram.org/bots/samples> (diakses Okt 20, 2020).
- [14] Joha, "Perancangan Sistem Komunikasi VoIP (Voice Over Internet Protocol) Berbasis SIP Dengan Menggunakan Metode PPDIIO Pada PT. Aplikasinusa Lintasarta Medan," *Tek. Inf.*, vol. 1, no. 1, hal. 40–46, 2019.