

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan Tinjauan pustaka di bawah ini memuat beberapa referensi yang di jadikan acuan penelitian, referensi tersebut adalah sebagai berikut:

Penelitian yang di lakukan David Karismata W. Tion (2016) membuat Bot Telegram untuk *Remote Monitoring* pada *server*. Aplikasi ini berfungsi untuk mengawasi kinerja dari *server* serta memberikan notifikasi kepada *Administrator* jika terjadi masalah.

Penelitian yang di lakukan Gde Sastrawangsa (2017) melakukan penelitian mengenai pemanfaatan Telegram yang diintegrasikan dengan Sistem Informasi Akademik. Sistem ini berfungsi untuk memberikan layanan dan informasi akademik melalui Telegram.

Penelitian yang di lakukan Gerhard Felix Sitorus (2019) melakukan penelitian tentang membuat bot menggunakan metode *webhooks* untuk pemesanan rental mobil dengan memanfaatkan telegram dan *website*.

Penelitian yang di lakukan Tomi Bong (2017) melakukan penelitian mengenai pemanfaatan Telegram *Bot* untuk mengakses portal akademik. Sistem ini berfungsi untuk memberikan informasi akademik melalui Telegram *Bot*.

Penelitian yang di lakukan Raga Nufusula dan Ajib Susanto (2018) membuat Chat Bot untuk server pulsa menggunakan Telegram API. Aplikasi ini berguna untuk menggantikan SMS dengan Bot Telegram untuk isi ulang pulsa prabayar.

Penelitian yang dilakukan Yoga Dwi Jatmika (2018) membuat aplikasi *E-Commerce* dengan menggunakan *framework Codeigniter* serta sebagai teknologi basisdata. Aplikasi ini mengolah pemesanan bahan bangunan pada U.D. Karya Utama.

Beberapa penelitian yang telah disebutkan di atas dapat dijadikan sebagai acuan dalam melakukan penelitian ini. Beberapa kelebihan dan kekurangan yang terdapat dalam penelitian sebelumnya akan di kaji dan di lakukan perbaikan, sehingga didapat hasil yang lebih baik. Berdasarkan hasil penelitian yang telah dipaparkan di atas, berikut ringkasan penelitian yang disajikan dalam Tabel 2.1.

Tabel Perbandingan

Tabel 2.1 Perbandingan Penelitian

Parameter Penulis	Judul	Metode / Teknologi	Bahasa Pemrograman	Studi Kasus
David Karismata W. Tion (2016)	Perancangan Bot untuk Remote Monitoring pada menggunakan Telegram Bot API	Telegram <i>Bot API</i> , <i>PPDIOO</i>	<i>Python</i>	Remote Monitoring Server
Gde Sastrawangsa (2017)	Pemanfaatan Telegram Untuk Automatisasi Layanan Dan Informasi Mahasiswa Dalam Konsep Smart Campus	Telegram <i>Bot API</i> , Webhook	<i>PHP, SQL</i>	Sistem Informasi Akademik

Gerhard Felix Sitorus (2019)	Pembuatan Telegram Bot Menggunakan Metode Webhook Untuk Pemesanan Rental Mobil	Telegram <i>Bot API,</i> <i>Webhook,</i> <i>Framework</i> <i>Codeigniter</i>	<i>PHP, SQL</i>	Rental Mobil
Tomi Bong (2017)	Pembangunan Aplikasi Media Informasi Mahasiswa Berbasis Bot Telegram	Telegram <i>Bot API,</i> <i>Waterfall</i>	<i>PHP</i>	Media Informasi Mahasiswa
Raga Nufusula, Ajib Susanto (2018)	Rancang Bangun Chat Bot Pada <i>Server</i> Pulsa Menggunakan Telegram Bot API	Telegram <i>Bot API,</i> <i>Long Polling</i>	<i>PHP, SQL</i>	<i>Server</i> Isi Ulang Pulsa
Yoga Dwi Jatmika (2018)	Pembangunan Website E-Commerce pada UD. Karya Utama	<i>Framework</i> <i>Codeigniter</i>	<i>PHP, SQL</i>	Toko Bangunan
Yudhi Prasetyo (2021)	Sistem Informasi Jadwal Praktek Dokter dengan Telegram Menggunakan Metode Webhook	Telegram <i>Bot API,</i> <i>Webhook,</i> <i>Framework</i> <i>Codeigniter,</i> <i>Waterfall</i>	<i>PHP, SQL</i>	Sistem Informasi Jadwal Praktek Dokter

2.2 Dasar Teori

2.2.1 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Perkembangan sistem informasi telah menyebabkan terjadinya perubahan yang cukup signifikan dalam pola pengambilan keputusan yang dilakukan oleh manajemen baik pada tingkat operasional.

Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya. Definisi sistem informasi dalam bukunya Abdul Kadir yang berjudul *Pengenalan Sistem Informasi*, yaitu: “sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan”. (Kadir, 2003:11)

2.2.2 Telegram Bot

Menurut Cokrojoyo (2017) Telegram menyediakan 2 bentuk API, API yang pertama adalah klien IM (Instant Message) Telegram, yang berarti semua orang dapat menjadi pengembang klien IM Telegram jika diinginkan. Ini berarti jika

seseorang ingin mengembangkan Telegram versi mereka sendiri mereka tidak harus memulai semua dari awal lagi. Telegram menyediakan *source code* yang mereka gunakan saat ini. Tipe API yang kedua adalah Telegram *Bot API*. API jenis kedua ini memungkinkan siapa saja untuk membuat bot yang akan membalas semua penggunaannya jika mengirimkan pesan perintah yang dapat diterima oleh Bot tersebut. Layanan ini masih hanya tersedia bagi pengguna yang menggunakan aplikasi Telegram saja. Sehingga pengguna yang ingin menggunakan Bot harus terlebih dahulu memiliki akun Telegram. Bot juga dapat dikembangkan oleh siapa saja.

Bot adalah aplikasi pihak ketiga yang berjalan di dalam Telegram. Pengguna dapat berinteraksi dengan bot dengan mengirimi mereka pesan, perintah, dan permintaan sebaris. Anda mengontrol bot Anda menggunakan permintaan HTTPS (*HyperText Transport Protocol Secure*) ke API bot. Bot API adalah antarmuka berbasis HTTP yang dibuat untuk pengembang yang tertarik membangun bot untuk telegram. API ini memungkinkan pengguna untuk menghubungkan bot ke sistem telegram. Telegram bots adalah akun khusus yang tidak memerlukan nomor telepon tambahan untuk mengatur. Akun ini berfungsi sebagai antarmuka untuk kode yang berjalan di suatu tempat di server pengguna.

Untuk menggunakan ini, pengguna tidak perlu tahu tentang bagaimana protokol enkripsi MTProto telegram bekerja, server perantara telegram akan menangani semua enkripsi dan komunikasi dengan API telegram untuk pengguna. Pengguna berkomunikasi dengan server ini melalui antarmuka HTTPS sederhana yang menawarkan versi sederhana dari telegram API. Ada beberapa metode yang

dapat digunakan untuk merancang sebuah Bot di Telegram. Beberapa diantaranya adalah : sendMessage, forwardMessage, sendPhoto, sendAudio, sendDocument, sendSticker, sendVideo, sendVoice, sendLocation, sendVenue, getUserProfilePhotos, sendChatAction, sendContact, unbanChatMember, kickChatMember, getFile, leaveChat, getChat, getChatAdministrator, getChatMember. Bot juga dapat menggunakan custom keyboard untuk penggunaanya. Hal ini akan mempermudah interaksi antara bot dan penggunaanya. Semua dasar pengiriman data yang digunakan oleh server Telegram akan menggunakan JSON, sehingga pengembang bot harus juga menggunakan bentuk data JSON. Bot Telegram tidak terbatas oleh bahasa pemrograman. Hampir semua bahasa pemrograman bisa digunakan untuk merancang suatu bot. Telegram juga menyediakan contoh bot yang menggunakan berbagai bahasa pemrograman. Fungsi dari telegram bot bermacam – macam tergantung dari kebutuhan masyarakat. Seperti sebagai alat untuk melakukan monitoring, alat untuk mengendalikan perintah tertentu serta bisa juga dijadikan sebagai alarm atau pengingat sesuatu. Selain itu aplikasi telegram juga sangat fleksibel karena dapat diakses melalui hampir semua perangkat baik android, ios maupun windows bahkan telegram juga memiliki telegram web yang dapat di akses melalui browser.

2.2.3 REST API

REST merupakan filosofi desain yang mendorong kita untuk menggunakan protokol dan fitur yang sudah ada pada web untuk memetakan permintaan terhadap sumber daya pada berbagai macam representasi dan manipulasi data di Internet (Scribner dan Seely, 2009).

REST adalah gaya arsitektural yang memiliki aturan seperti antar muka yang seragam, sehingga jika aturan tersebut diterapkan pada web services akan dapat memaksimalkan kinerja web services terutama pada performa, skalabilitas, dan kemudahan untuk dimodifikasi. Pada arsitektur REST, data dan fungsi dianggap sebagai sumber daya yang dapat diakses lewat Uniform Resource Identifier (URI), biasanya berupa tautan pada web.

REST menggunakan protokol HTTP yang bersifat stateless. Perintah HTTP yang bisa digunakan adalah fungsi GET, POST, PUT atau DELETE. Hasil yang dikirimkan dari server biasanya dalam bentuk format XML atau JSON sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing pada sisi client.

Dalam penerapannya, REST lebih banyak digunakan untuk web service yang berorientasi pada sumber daya. Maksud orientasi pada sumber daya adalah orientasi yang menyediakan sumber daya sebagai layanannya dan bukan kumpulan-kumpulan dari aktifitas yang mengolah sumber daya itu. Bentuk web service menggunakan REST style sangat cocok digunakan sebagai back-end dari aplikasi berbasis mobile karena cara aksesnya yang mudah dan hasil data yang dikirimkan berformat JSON sehingga ukuran file menjadi lebih kecil.

2.2.4 Metode Waterfall

Metode *Waterfall* merupakan pendekatan SDLC paling awal yang digunakan untuk pengembangan perangkat lunak. Urutan dalam Metode *Waterfall* bersifat serial yang dimulai dari proses perencanaan, analisa, desain, dan implementasi pada sistem.

Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh karena itu di sebut *waterfall* (Air Terjun).

Ada lima tahapan pada Metode *Waterfall*, yakni *Requirements Analysis and Definition*, *System and Software Design*, *Implementation and Unit Testing*, *Integration and System Testing*, dan *Operationa and Maintenance*. (Ian Sommerville, 2011)

2.2.5 Metode *Webhook*

Salah satu metode yang dapat digunakan untuk membuat Telegram ialah *webhook* dengan *set webhook*. Metode ini digunakan untuk menentukan *URL* dan menerima pembaruan yang masuk melalui *webhook* keluar. Setiap kali ada pembaruan untuk *Bot*, Telegram akan mengirimkan permintaan *POST HTTPS* ke *URL* yang ditentukan, yang berisi pembaruan dengan bentuk *JSON*. Jika permintaan tidak berhasil, telegram akan berhenti mengirim *request* setelah upaya yang wajar. Sebaliknya, respon *true* jika *request* dinyatakan sukses.

Untuk memastikan bahwa permintaan Webhook adalah benar berasal dari Telegram, digunakan jalur rahasia pada URL, mis. <https://www.example.com/<token>>. Karena token bot hanya diketahui oleh pembuat *Bot* itu sendiri, maka *request* dapat diyakini benar berasal dari Telegram (The Telegram Team, 2015). Parameter yang dapat digunakan untuk membuat Telegram *Bot* dengan metode *Webhook* dapat dilihat pada tabel.

Tabel 2.2 Parameter Telegram Bot Metode Webhook

Parameter	Tipe data	Dibutuhkan	Deskripsi
<i>url</i>	<i>String</i>	Ya	<i>URL HTTPS</i> untuk mengirim pembaruan ke bot.
<i>Certificate</i>	<i>InputFile</i>	Opsional	Digunakan untuk mengunggah sertifikat kunci publik sehingga sertifikat root yang digunakan dapat diperiksa.
<i>max_conne ction</i>	<i>Integer</i>	Opsional	Jumlah maksimum koneksi <i>HTTPS</i> simultan yang diizinkan ke <i>webhooks</i> untuk pengiriman pembaruan, 1-100. Standarnya menjadi 40.
<i>allowed_up date</i>	<i>Array of String</i>	Opsional	Daftar jenis pembaruan yang diinginkan untuk diterima bot. Jika tidak ditentukan, pengaturan sebelumnya akan digunakan.

2.2.6 Web Server

Web server adalah perangkat lunak yang menyediakan layanan permintaan data dengan protokol *HTTP* atau *HTTPS* yang diakses melalui aplikasi tertentu seperti *web browser*. *Web* merespon permintaan dengan mengirimkan kembali konten berupa gambar, tulisan, atau lainnya dan kemudian ditampilkan melalui browser (Singh & Kumar, 2011).

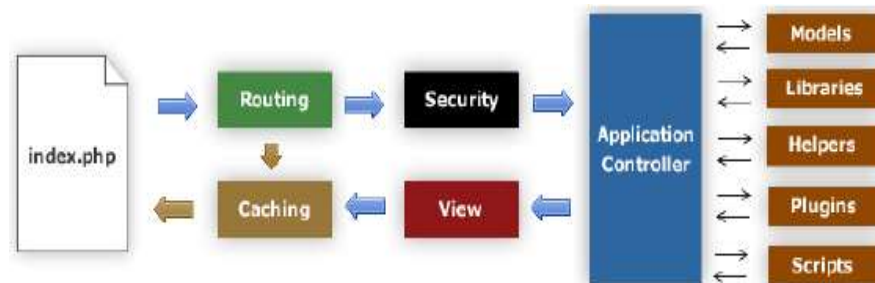
2.2.7 Codeigniter

Codeigniter merupakan *open source* berbasis MVC (*Model View Controller*).

Untuk membangun aplikasi web dinamis dengan cepat dan mudah. *Codeigniter* memiliki desain dan struktur *file* yang sederhana, didukung dengan dokumentasi yang lengkap sehingga *framework* ini lebih mudah dipelajari.

Codeigniter ini memungkinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan. Artinya bahwa *Codeigniter* masih memberi kebebasan kepada para pengembang untuk menulis bagian – bagian kode tertentu di dalam aplikasi menggunakan cara konvensional atau dengan sintak umum di dalam *PHP*, tidak harus menggunakan aturan penulisan kode di *Codeigniter* (Septian, 2011).

Adapun alur dari aplikasi yang ditulis menggunakan *Codeigniter* adalah sebagai berikut:



Gambar 2.1 Alur Penggunaan Codeigniter

File *index.php*, berfungsi sebagai *front controller*, menginisialisasi *resource* utama yang dibutuhkan untuk menjalankan *Codeigniter*.

1. *Router*, memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan.
2. Jika *file cache* ada, dikirim langsung ke browser, melewati eksekusi sistem normal.

3. Keamanan, sebelum *controller* aplikasi dimuat, HTTP *request* dan setiap data pengguna yang dikirim dan disaring dahulu untuk keamanan.
4. *Controller* memuat model, *library* utama, *helper* dan setiap *resource* lainnya yang diperlukan untuk memproses permintaan khusus.
5. *View*, proses render kemudian dikirim ke web browser agar dapat dilihat. Jika caching diaktifkan, *view* dicache terlebih dahulu sehingga pada permintaan berikutnya dapat dilayani.

2.2.8 Apache

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows, serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah (Wardani, 2013).

2.2.9 Hypertext Preprocessor (PHP)

PHP singkatan (PHP Hypertext Preprocessor) adalah bahasa server-side scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Server-side scripting adalah sintak dan perintah – perintah yang diberikan akan sepenuhnya dijalankan di server tapi disertakan pada dokumen HTML. (Bimo Sunarfrihantono, 2002).

2.2.10 *MySQL*

MySQL tergolong sebagai *DBMS (Database Management System)*. Perangkat lunak ini bermanfaat untuk mengelola data yang sangat fleksibel dan juga cepat. *MySQL* banyak dipakai untuk penanganan database, selain handal juga bersifat *open source* (Abdul Kadir, 2009).

2.2.11 *Unified Modeling Language (UML)*

Menurut Suhendar dan Gunadi (2002), Unified Modeling Language (UML) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi dan mendokumentasikan artifact dari sistem software, untuk memodelkan bisnis dan sistem nonsoftware lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks.

Artifact adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa perangkat lunak. Artifact dapat berupa model, deskripsi, atau perangkat lunak. Tujuan utama UML diantaranya untuk :

- a. Memberikan model yang siap pakai, bahasa permodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
- b. Memberikan bahasa permodelan yang bebas dari berbagai bahasa pemrograman dan praktik rekayasa.
- c. Menyatukan praktik-praktik terbaik yang terdapat dalam permodelan.