

BAB II

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Aplikasi

Menurut Hendrayudi (2009, 143), “Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus)”.

2.1.2 Data

Sumber informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal *datum* atau *item*. Menurut McLeod dalam bukunya Yakub (2012, 5), “data adalah deskripsi kenyataan yang menggambarkan adanya suatu kejadian (*event*), data terdiri dari fakta (*fact*) dan angka yang secara relatif tidak berarti bagi pemakai”.

Menurut Anggraeni dan Irviani (2017, 12-13), “Data adalah deskripsi tentang benda, kejadian, aktivitas, dan transaksi, yang mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai”.

Tabel yang menyuguhkan data yang bermakna :

Tabel 2.1 Contoh Tabel Data

Waktu	Suhu
6.30	27
6.32	28
6.34	27

Terdapat 5 (lima) tipe data yaitu sebagai berikut:

- a. Data yang terformat, adalah data dengan suatu format tertentu. Misalnya data yang menyatakan tanggal atau jam, atau menyatakan nilai mata uang.
- b. Teks, adalah sederetan huruf, angka, dan simbol-simbol khusus (misalnya + dan \$) yang kombinasinya tidak tergantung pada masing-masing *item* secara individual. Contoh teks adalah artikel koran.
- c. Citra (*image*), adalah data dalam bentuk gambar. Citra dapat berupa foto, grafik, hasil *rontgen*, dan tanda tangan, dan lain-lain.
- d. Audio, adalah data dalam bentuk suara. Contoh data audio: instrumen musik, suara orang/suara binatang, gemericik air, dan detak jantung.
- e. Video, menyatakan data dalam bentuk sejumlah gambar yang bergerak dan bisa digunakan untuk mengabadikan suatu kejadian/aktivitas.

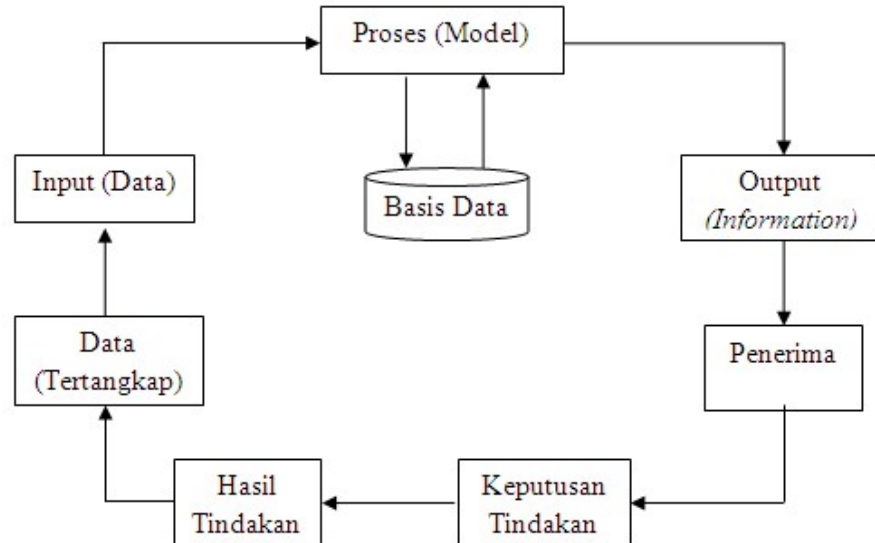
(Hutahaean 2014) Pengolahan data adalah serangkaian operasi atas informasi yang direncanakan guna mencapai tujuan atau hasil yang diinginkan. Ada delapan unsur pokok pengolahan data, yaitu membaca, menulis atau mengetik, mencatat atau mencetak, menyortir, menyampaikan atau memindahkan, menghitung, membandingkan, dan menyimpan.

2.1.3 Informasi

(Anggraeni dan Irviani 2017) Informasi adalah sekumpulan data/fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima. Data yang telah diolah menjadi sesuatu yang berguna bagi penerima yaitu dapat memberikan keterangan atau pengetahuan. Dengan demikian yang menjadi sumber informasi adalah data. Informasi dapat juga dikatakan sebuah pengetahuan yang diperoleh dari pembelajaran, pengalaman, atau instruksi.

(Hutahaean 2014) Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang. Fungsi utama informasi yaitu menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi, karena informasi berguna memberikan gambaran tentang suatu permasalahan sehingga pengambil keputusan dapat menentukan keputusan lebih cepat. Informasi juga memberikan standar, aturan maupun indikator bagi pengambil keputusan.

Data yang diolah untuk menghasilkan informasi menggunakan model proses yang tertentu. Misalkan suhu dalam fahrenheit diubah ke celcius. Dalam hal ini digunakan model matematik berupa rumus konversi dari derajat fahrenheit menjadi satuan derajat celcius. Data yang diolah melalui suatu model menjadi informasi, kemudian penerima menerima informasi tersebut, yang berarti menghasilkan keputusan dan melakukan tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali melalui suatu model dan seterusnya yang disebut dengan siklus informasi (*information cycle*). Siklus ini juga disebut dengan siklus pengolahan data (*data processing cycles*).



Gambar 2.1 Siklus Informasi

(sumber: Konsep Sistem Informasi. 2014:11)

Nilai informasi ditentukan oleh dua hal yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan lebih bernilai jika manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

2.2 Teori Khusus

2.2.1 Pusat Informasi Mahasiswa

Berikut ini adalah arti perkata dari “Pusat Informasi Mahasiswa” menurut Kamus Besar Bahasa Indonesia (KBBI):

- a. Pusat : tempat yang letaknya di bagian tengah; titik yang di tengah-tengah benar (dalam bulatan bola, lingkaran, dan sebagainya); pusat; pokok pangkal atau yang menjadi pempunan (berbagai-bagai urusan, hal, dan sebagainya); orang yang membawahkan berbagai bagian; orang yang menjadi pempunan dari bagian-bagian.

- b. Informasi : penerangan; pemberitahuan; kabar atau berita tentang sesuatu; keseluruhan makna yang menunjang amanat yang terlihat dalam bagian-bagian amanat itu.
- c. Mahasiswa : orang yang belajar di perguruan tinggi.

Dengan demikian dapat ditarik kesimpulan bahwa Pusat Informasi Mahasiswa adalah suatu pusat, suatu wadah yang digunakan untuk memberi informasi lengkap kepada orang yang belajar di perguruan tinggi atau yang berhubungan dengan mahasiswa.

2.2.2 *Bot*

(Christensson 2014) mengungkapkan bahwa: “*A Bot (short for "roBot") is an automated program that runs over the Internet. Some Bots run automatically, while others only execute commands when they receive specific input. There are many different types of Bots, but some common examples include web crawlers, chat room Bots, and malicious Bots*”.

Sebuah *Bot* (singkatan dari “*roBot*”) adalah program otomatis yang berjalan melalui internet. Beberapa *Bot* berjalan secara otomatis, sementara yang lainnya hanya menjalankan perintah ketika mereka menerima masukan spesifik. Ada berbagai banyak jenis *Bot*, tapi beberapa contoh umum seperti *web crawlers*, *chat room Bots*, dan *malicious Bots*.

a. *Web Crawlers*

(Christensson 2014) mengungkapkan bahwa: “*Web crawlers are used by search engines to scan websites on a regular basis. These Bots "crawl" websites by following the links on each page. The crawler saves the contents of each page in the search index. By using complex*

algorithms, search engines can display the most relevant pages discovered by web crawlers for specific search queries”.

Web crawlers (perayap web) digunakan oleh mesin pencari untuk memindai situs web secara teratur. *Bots* ini “merayapi” situs web dengan mengikuti tautan di setiap halaman. Perayap ini menyimpan konten setiap halaman dalam indeks pencarian. Dengan menggunakan algoritma kompleks, mesin pencari dapat menampilkan halaman yang paling relevan yang ditemukan oleh perayap web untuk *query* pencarian tertentu.

b. *Chat Room Bots*

(Christensson 2014) mengungkapkan bahwa: “*Chat Bots were one of the first types of automated programs to be called "Bots" and became popular in the 1990s, with the rise of online chatrooms. These Bots are scripts that look for certain text patterns submitted by chat room participants and respond with automated actions. For example, a chat Bot might warn a user if his or language is inappropriate. If the user does not heed the warning, the Bot might kick the user from the channel and may even block the user from returning. A more advanced type of chat Bot, called a "chatterBot" can respond to messages in plain English, appearing to be an actual person. Both types of chat Bots are used for chatroom moderation, which eliminates the need for an individual to monitor individual chatrooms”.*

Chat Bots adalah salah satu jenis pertama dari program otomatis yang disebut “*Bot*” dan menjadi populer pada tahun 1990-an, dengan munculnya *online chatrooms* (ruang obrolan *online*). *Bot* ini adalah *scripts* yang mencari pola teks tertentu yang dikirim oleh peserta/pengguna

chatroom dan merespons dengan otomatis. Sebagai contoh, *Bot* mungkin memperingatkan pengguna jika bahasanya tidak tepat. Jika pengguna tidak menginginkan peringatan tersebut, *Bot* mungkin mengeluarkan pengguna dari *channel* dan bahkan dapat memblokir pengguna untuk kembali. Jenis *chat Bots* yang lebih canggih yang disebut “*chatterBot*” dapat merespons pesan dalam bahasa Inggris biasa, yang tampak sebagai orang yang sebenarnya. Kedua jenis *chat Bots* digunakan untuk moderasi *chatroom*, yang menghilangkan kebutuhan individu untuk memantau *chatroom* individual.

(Zumstein dan Hundertmark 2017) mengungkapkan bahwa: “*The word “chatBot” consists of the terms “chat” and “roBot”. Originally, the term chatBot was used for a computer program, which simulates human language with the aid of a text-based dialogue system. ChatBots contain a text input and output mask, which allows mobile users to communicate with the software behind them, giving them the feeling of chatting with a real person*”.

Kata “*chatBot*” terdiri dari istilah “*chat*” dan “*roBot*”. Awalnya, istilah *chatBot* digunakan untuk program komputer, yang menstimulasikan bahasa manusia dengan bantuan sistem dialog berbasis teks. *ChatBot* berisi teks masukan dan keluaran, yang memungkinkan pengguna seluler berkomunikasi dengan perangkat lunak di belakangnya, memberi mereka perasaan mengobrol dengan orang sungguhan.

c. *Malicious Bots*

(Christensson 2014) mengungkapkan bahwa: “*While most Bots are used for productive purposes, some are considered malware, since they*

perform undesirable functions. For example, spamBots capture email addresses from website contact forms, address books, and email programs, then add them to a spam mailing list. Site scrapers download entire websites, enabling unauthorized duplication of a website's contents. DoS Bots send automated requests to websites, making them unresponsive. Botnets, which consist of many Bots working together, may be used to gain unauthorized access to computer systems and infect computers with viruses”.

Sementara kebanyakan *Bot* digunakan untuk tujuan produktif, beberapa dianggap *malware*, karena mereka melakukan fungsi yang tidak diinginkan. Misalnya, *spamBots* menangkap alamat *email* dari formulir kontak *website*, buku alamat, dan program *email*, lalu menambahkan mereka ke daftar *spam mailing*. *Site scrapers* (pengeruk situs) mengunduh keseluruhan situs web, memungkinkan duplikasi situs web yang tidak sah. *DoS Bots* mengirim permintaan otomatis ke situs web, membuat mereka tidak responsif. *Botnet*, yang terdiri dari banyak *Bot* yang bekerja bersama, dapat digunakan untuk mendapatkan akses tidak sah ke sistem komputer dan menginfeksi komputer dengan virus.

2.2.3 API (Application Programming Interface)

(Jacobson, Brail and Woods 2011) mengungkapkan bahwa: “*API stands for application programming interface. An API can provide a hook for colleagues, partners, or third-party developers to access data and services to build applications such as iPhone apps quickly. The Twitter and Facebook APIs are famous examples. There are APIs that are open to any developer,*

APIs that are open only to partners, and APIs that are used internally to help run the business better and facilitate collaboration between teams”.

API adalah singkatan dari antarmuka pemrograman aplikasi. Sebuah API dapat memberikan pengait bagi rekan, mitra, atau pengembang pihak ketiga untuk mengakses data dan layanan untuk membangun aplikasi seperti aplikasi *iPhone* dengan cepat. API *Twitter* dan *Facebook* adalah contoh terkenal. Ada API yang terbuka untuk semua pengembang, API yang hanya terbuka untuk mitra, dan API yang digunakan secara internal untuk membantu menjalankan bisnis dengan lebih baik dan memfasilitasi kolaborasi antar tim.

Sedangkan (Christensson 2016) mengungkapkan bahwa: “*Stands for "Application Programming Interface." An API is a set of commands, functions, protocols, and objects that programmers can use to create software or interact with an external system. It provides developers with standard commands for performing common operations so they do not have to write the code from scratch”.*

Singkatan untuk “*Application Programming Interface*”. API adalah sekumpulan perintah, fungsi, protokol, dan objek yang dapat digunakan *programmer* untuk membuat perangkat lunak atau berinteraksi dengan sistem eksternal. API menyediakan pengembangan dengan perintah standar untuk melakukan operasi umum sehingga mereka tidak harus menulis kode dari awal.

2.2.4 Telegram Messenger

(Telegram Team 2013) *Telegram* adalah aplikasi pesan instan berbasis *cloud* yang berfokus pada kecepatan dan keamanan. *Telegram* dapat digunakan di semua perangkat pada saat yang sama, semua pesan disinkronkan dengan

mulus di sejumlah *smartphone*, *tablet* atau komputer. Pengguna dapat mengirim pesan, foto, video, dan berkas jenis apapun hingga 1,5 *Gigabyte*, serta membuat grup hingga 100.000 orang atau saluran untuk disiarkan ke pengguna tak terbatas.

Telegram memiliki API (*Application Programming Interface*) yang terbuka dan gratis, salah satunya *Bot API*, antarmuka berbasis HTTP (*HyperText Transfer Protocol*) yang dibuat untuk pengembang membuat *Bot* untuk *Telegram*.

2.2.5 Telegram Bot API (*Application Programming Interface*)

(Risanty dan Sopiyan 2017) *Telegram Bot API* (*Application Programming Interface*) adalah sebuah teknologi *open source* yang disediakan oleh *Telegram* untuk membangun aplikasi *Bot Telegram* bagi para pengembang. *Bot API* ini merupakan *interface* berbasis HTTP untuk menghubungkan *Bot* yang dikembangkan oleh para pengembang dengan sistem *Telegram*.

Bot Telegram merupakan sebuah akun khusus yang tidak memerlukan nomor telepon. Akun ini berfungsi sebagai *interface* untuk menjalankan *code* yang sudah dibangun. Untuk keamanan data, *server* perantara pada *Telegram* akan menangani semua enkripsi dan komunikasi dengan *Bot API*, sehingga para pengembang tidak perlu mengetahui bagaimana protokol enkripsi MTProto pada *server Telegram* bekerja.

Terdapat beberapa metode yang digunakan untuk merancang sebuah *Bot* di *Telegram Messenger*. Beberapa diantaranya adalah `sendMessage`, `sendPhoto`, `sendDocument`, `sendVideo`, `sendLocation`, `editMessageText`.

Semua *query* ke *Telegram Bot* API harus melalui HTTPS dan disajikan dalam bentuk `https://api.telegram.org/bot<token>/NAMA_METODE`. Contohnya: `https://api.telegram.org/bot123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11/getMe`.

Bot Telegram juga mendukung pengiriman text berformat seperti teks tebal, teks miring atau URL, dengan menggunakan sintaks dibawah ini.

`*bold text*`

`_italic text_`

`[inline URL](http://www.example.com/)`

`[inline mention of a user](tg://user?id=123456789)`

``inline fixed-width code``

````block_language`

`pre-formatted fixed-width code block`

`````

2.2.6 *Regular Expression*

(Levithan dan Goyvaerts 2012) mengungkapkan bahwa: “*a regular expression is a specific kind of text pattern that you can use with many modern applications and programming languages. you can use them to verify whether input fits into the text pattern, to find text that matches the pattern within a larger body of text, to replace text matching the pattern with other text or rearranged bits of the matched text, to split a block of text into a list of subtexts*”.

Regular Expression adalah jenis pola teks spesifik yang dapat anda gunakan dengan banyak aplikasi modern dan bahasa pemrograman. Anda dapat menggunakannya untuk memverifikasi apakah masukan (*input*) sesuai dengan pola teks, untuk menemukan teks yang cocok dengan pola dalam badan teks yang lebih besar, untuk mengganti teks yang cocok dengan pola dengan teks lain atau bagian yang disusun kembali dari teks yang cocok, untuk membagi blok teks ke dalam daftar subteks.

(Nagy 2018) *Regular Expression* digunakan untuk:

- a. Menguji apakah *string* cocok dengan pola pencarian.
- b. Menemukan beberapa karakter dalam *string*.
- c. Mengganti *substring* dalam *string* yang cocok dengan *Regular Expression*.
- d. Memproses dan memformat masukan pengguna.
- e. Mengekstrak informasi dari log *server*, *file* konfigurasi, dan *file* teks.
- f. Memvalidasi masukan pengguna dalam aplikasi web dan di terminal.

2.2.7 Fulltext Search Boolean Mode

Menurut Paliwahet, Sukarsa dan Putra (2017, 146), “*Fulltext Search Boolean Mode* merupakan fitur yang digunakan untuk proses pencocokan. Pencocokan yang dilakukan berbeda dengan pencocokan biasa. Pencocokan dengan metode ini dapat menghasilkan hasil yang paling relevan. Fitur ini dapat diterapkan pada *engine* InnODB dan MyISAM pada MySQL versi 5.6. Kelebihan yang dimiliki *Fulltext Search Boolean Mode* yaitu dapat mengimplementasikan operator dalam pencarian”.

Tabel 2.2 Operator *Fulltext Search Boolean Mode*

| Operator | Deskripsi |
|-------------|--|
| + | Operator tambah menunjukkan bahwa kata ini harus ada di setiap baris yang ingin dicocokkan. <i>Engine</i> InnoDB hanya mendukung tanda tambah pada awal kata. |
| - | Operator minus menunjukkan bahwa kata ini tidak boleh ada di salah satu baris yang ingin dicocokkan. InnoDB hanya mendukung tanda minus pada awal kata. |
| No operator | Tanpa operator merupakan konfigurasi <i>default</i> pada MATCH() ... AGAINST() |
| @distance | Operator ini bekerja pada tabel InnoDB saja. Operator ini menguji apakah dua kata atau lebih semuanya dimulai dalam jarak yang ditentukan satu sama lain, diukur dengan kata-kata. Tentukan kata-kata pencarian dalam <i>string</i> kutipan ganda sebelum operator @distance, misalnya MATCH (col1) AGAINST (“word1 word2 word3” @ 8’ IN BOOLEAN MODE) |
| > < | Kedua operator ini digunakan untuk mengubah kontribusi kata ke nilai relevansi yang ditugaskan berturut-turut. Operator > meningkatkan kontribusi dan operator < menurunkannya. |
| ~ | Operator negasi, menyebabkan kontribusi kata tersebut ke relevansi baris menjadi negatif. Ini berguna menandai kata-kata “ <i>noise</i> ”. Baris yang berisi kata tersebut dinilai lebih rendah dari yang lain, namun tidak dikecualikan sama sekali, seperti halnya dengan operator -. |
| * | Operator * berfungsi sebagai operator pemotongan (atau <i>wildcard</i>). Kata cocok jika mereka dimulai dengan kata yang didahului operator *. |
| “ | Operator tanda petik dua yaitu untuk mencari kata atau frasa yang benar-benar sama dengan masukan. |

Tabel 2.2 merupakan operator yang dapat digunakan pada *Boolean Mode*. Penerapan *Boolean Mode Fulltext* dapat diterapkan dalam suatu tabel

yang ingin dicocokkan. Penggunaan fitur *Fulltext Search Boolean Mode* ini dapat dilakukan dengan cara menambahkan indeks *FULLTEXT* pada kolom yang ingin dijadikan patokan pencocokan.

```
CREATE TABLE `tb_pattern` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `pattern` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`),
  FULLTEXT KEY `pattern` (`pattern`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Kode program di atas merupakan kode program untuk pembuatan tabel yang berisikan indeks *FULLTEXT*. Kolom yang mendapatkan indeks tersebut dapat digunakan dalam pencocokan *Boolean Mode*. Penerapan *Boolean Mode* ini dapat digunakan dengan menggunakan kombinasi *SELECT*, *MATCH* dan *AGAINST* pada sintaks *MySQL*.

```
SELECT * FROM tb_pattern WHERE MATCH(pattern) AGAINST('lokasi unud' IN BOOLEAN MODE);
```

Kode program di atas merupakan kode program sederhana dalam penggunaan *Boolean Mode*. Kode program tersebut dapat sedikit dimodifikasi sehingga nilai relevansi terlihat pada setiap *pattern*. Berikut ini merupakan kode program untuk menampilkan relevansi pada setiap *pattern chatbot*.

```
SELECT id, pattern, MATCH(pattern)
AGAINST('lokasi unud' IN BOOLEAN MODE) AS relevansi
FROM tb_pattern ORDER BY(relevansi) DESC;
```

Hasil yang akan ditunjukkan dari kode program di atas yaitu id *pattern*, *pattern chatbot* dan nilai relevansi *pattern* dengan kalimat masukan. Pencocokan *Boolean Mode* menerapkan rumus yang diterapkan berbasis dari algoritma BM25 dan TF-IDF. Fungsi *Boolean Mode* dapat dilihat pada persamaan (1).

$$(1) \ \{rank\} = \{TF\} * \{IDF\} * \{IDF\}$$

Dengan:

TF : *Term Frequency*, banyak munculnya kata dalam kalimat

IDF : *Inverse Document Frequency*

IDF dapat diketahui dengan jumlah dokumen dibagi dengan jumlah dokumen yang terkait dengan masukan yang diberikan. Persamaan (2) merupakan rumus untuk mendapatkan nilai IDF.

$$(2) \ \{IDF\} = \log_{10} \left(\frac{\{total_{records}\}}{\{matching_{records}\}} \right)$$

Persamaan 2 menunjukkan rumus untuk menentukan nilai IDF dari suatu dokumen atau data. *Boolean Mode* juga dapat diterapkan pada satu kata atau lebih yang menjadi patokan dalam pencarian.

$$(3) \ \{rank\} = \sum(\{TF\} * \{IDF\} * \{IDF\})$$

Persamaan 3 merupakan rumus untuk mencari relevansi data dengan pencarian lebih dari satu kata. Nilai relevansi didapat dengan menjumlahkan nilai relevansi pada setiap kata pencarian.

2.3 Teori Perancangan

2.3.1 PHP (PHP: *Hypertext Preprocessor*)

Menurut Sibero dalam bukunya Supono (2018, 3), “PHP (*Personal Home Page*) adalah pemrograman (*interpreter*) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”.

Menurut Anhar (2010, 3), “PHP singkatan dari PHP: *Hypertext Preprocessor*) yaitu bahasa pemrograman web *server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded Scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan”.

Ada beberapa alasan yang menjadi dasar pertimbangan mengapa menggunakan PHP:

- a. Mudah dipelajari, alasan tersebut menjadi salah satu alasan utama untuk menggunakan PHP. Pemula pun akan mampu untuk menjadi *web master* PHP.
- b. *Cross Platform*, artinya PHP dapat mudah diaplikasikan ke berbagai *platform OS (Operating System)* dan hampir semua *browser* juga mendukung PHP.
- c. *Open Source* atau gratis.

- d. PHP memiliki tingkat akses yang cepat.
- e. Didukung oleh beberapa macam *web server*, PHP mendukung beberapa *web server* seperti Apache, IIS, Lighttpd, Xitami, dan lain-lain.
- f. Mendukung *database*, PHP mendukung beberapa *database*, baik yang gratis maupun yang berbayar, seperti MySQL, PostgreSQL, mSQL, Informix, SQL Server, Oracle.

2.3.2 Basis Data (*Database*)

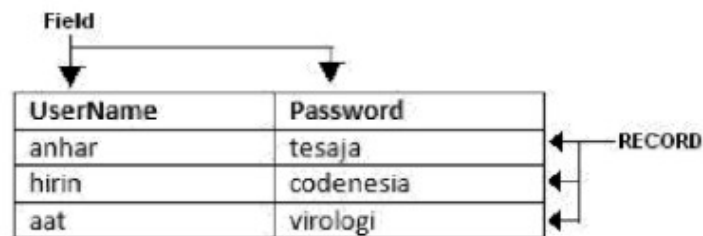
(Fathansyah 2015) Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Sebagai satu kesatuan istilah, Basis Data (*Database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

(Anhar 2010) Struktur *file* yang menyusun sebuah *database* adalah sebagai berikut:

- a. Data adalah satu satuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan di dalam suatu *file database*.
- b. *Record* adalah data yang isinya merupakan satu kesatuan seperti *Username* dan *Password*. Setiap keterangan yang mencakup *Username* dan *Password* dinamakan satu *record*. Setiap *record* diberi nomor urut yang disebut nomor *record* (*Record Number*).
- c. *Field* adalah sub bagian dari *Record*. Dari contoh isi *record* di atas, maka terdiri dari 2 *field*, yaitu: *field Username* dan *Password*.



Gambar 2.2 Tabel *Database Username dan Password*

(sumber: PHP & MySql Secara Otodidak. 2010:46)

2.3.3 MySQL

(Anhar 2010) MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL, dan lain-lain. MySQL merupakan DBMS yang *multithread*,

multi-user yang bersifat gratis di bawah lisensi GNU *General Public License* (GPL).

Beberapa kelebihan MySQL, antara lain:

- a. Dapat berjalan dengan stabil pada berbagai sistem operasi, seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris*, dan masih banyak lagi.
- b. Bersifat *open source*.
- c. Bersifat *multi-user*, dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
- d. MySQL memiliki kecepatan yang baik dalam menangani *query* (perintah SQL).
- e. MySQL memiliki beberapa lapisan keamanan, seperti level *subnet mask, nama host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta password yang terenkripsi.

2.3.4 *Framework*

Menurut Wardana (2010, 3), “*Framework* adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi *website*, kita harus mengikuti aturan dari *framework* tersebut”.

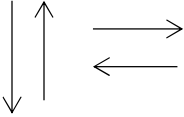




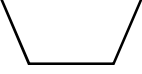
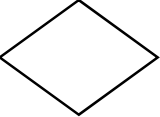

Menurut Pratama (2010, 10) mengatakan bahwa *framework* adalah rangka atau kerangka, arti istilah tersebut dalam dunia pemrograman adalah kumpulan kelas (*class*) dan fungsi (*function, method*) yang disusun secara sistematis berdasarkan kegunaan atau fungsi tertentu untuk mempermudah pembuatan atau pengembangan suatu aplikasi. Sebagian besar *framework* yang


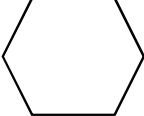




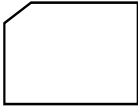

berredar saat ini dibangun berdasarkan konsep *Object-Oriented Programming*. Selain itu banyak manfaat yang didapat saat menggunakan *framework*. *Framework* menawarkan penghematan waktu kerja dalam penulisan kode dan pengaturan berkas-berkas kode. *Programmer* tidak perlu susah payah menulis kode dari awal untuk fungsi-fungsi yang sudah disediakan. Selain itu berkas kode dapat disusun secara sistematis sesuai dengan struktur yang ditawarkan *framework*.

2.3.5 *Flowchart*

Menurut Joe Purba (2018, 1) *Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan proses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah *flowchart* selesai disusun, selanjutnya pemrogram (*programmer*) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

Flowchart disusun dengan simbol-simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang dipakai antara lain:

| Nama Simbol <i>Flowchart</i> | Penjelasan |
|---|--|
| <p data-bbox="472 254 760 285"><i>Flow Direction Symbol</i></p>  | <p data-bbox="813 279 1344 415">Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga <i>connecting line</i>.</p> |
| <p data-bbox="500 464 732 495"><i>Terminator Symbol</i></p>  | <p data-bbox="824 506 1333 573">Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu kegiatan.</p> |
| <p data-bbox="505 659 727 690"><i>Connector Symbol</i></p>  | <p data-bbox="824 680 1328 783">Yaitu simbol untuk keluar-masuk atau penyambungan proses dalam lembar atau halaman yang sama.</p> |
| <p data-bbox="505 842 727 873"><i>Connector Symbol</i></p>  | <p data-bbox="808 875 1349 978">Yaitu simbol untuk keluar-masuk atau penyambungan proses pada lembar/ halaman yang berbeda.</p> |
| <p data-bbox="500 1041 732 1073"><i>Processing Symbol</i></p>  | <p data-bbox="808 1083 1344 1150">Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.</p> |
| <p data-bbox="456 1245 776 1276">Simbol <i>Manual Operating</i></p>  | <p data-bbox="808 1276 1344 1344">Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.</p> |
| <p data-bbox="516 1419 716 1451">Simbol <i>Decision</i></p>  | <p data-bbox="849 1472 1304 1539">Simbol pemilihan proses berdasarkan kondisi yang ada.</p> |
| <p data-bbox="480 1625 756 1656">Simbol <i>Input – Output</i></p>  | <p data-bbox="821 1646 1336 1749">Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.</p> |

| | |
|---|--|
| <p>Simbol Manual <i>Input</i></p>  | <p>Simbol untuk pemasukan data secara manual <i>on-line keyboard</i>.</p> |
| <p>Simbol <i>Preparation</i></p>  | <p>Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i>.</p> |
| <p>Simbol <i>Predefine</i> Proses</p>  | <p>Simbol untuk pelaksanaan suatu bagian (sub-program) atau <i>procedure</i>.</p> |
| <p>Simbol <i>Display</i></p>  | <p>Simbol yang menyatakan peralatan <i>output</i> yang digunakan yaitu layar, <i>plotter</i>, <i>printer</i> dan sebagainya.</p> |
| <p>Simbol <i>Disk and On-line Storage</i></p>  | <p>Simbol yang menyatakan <i>input</i> yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i>.</p> |
| <p>Simbol Magnetik Tape Unit</p>  | <p>Simbol yang menyatakan <i>input</i> berasal dari pita magnetik atau <i>output</i> disimpan ke pita magnetik.</p> |
| <p>Simbol <i>Punch Card</i></p>  | <p>Simbol yang menyatakan bahwa <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.</p> |
| <p>Simbol Dokumen</p>  | <p>Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.</p> |

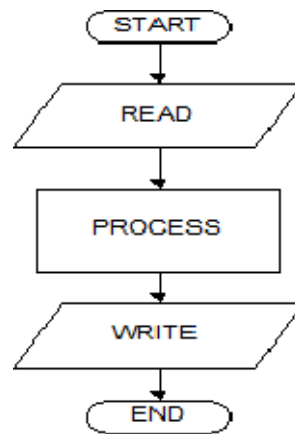
Dalam pembuatan *flowchart* tidak ada rumus atau patokan yang bersifat mutlak. Karena *flowchart* merupakan gambaran hasil pemikiran dalam menganalisa suatu masalah dengan komputer. Sehingga *flowchart* yang dihasilkan dapat bervariasi antara satu pemrogram dengan pemrogram lainnya.

Namun secara garis besar, setiap pengolahan selalu terdiri dari tiga bagian utama, yaitu:

- a. *Input* berupa bahan mentah
- b. Proses pengolahan
- c. *Output* berupa bahan jadi.

Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu;

- a. START: berisi instruksi untuk persiapan peralatan yang diperlukan sebelum menangani pemecahan masalah.
- b. READ: berisi instruksi untuk membaca data dari suatu peralatan *input*.
- c. PROCESS: berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- d. WRITE: berisi instruksi untuk merekam hasil kegiatan ke peralatan *output*.
- e. END: mengakhiri kegiatan pengolahan.



Gambar 2.3 Memperlihatkan *Flowchart* Dari Kegiatan Dasar Di atas.

(sumber:

https://www.academia.edu/6912277/PENGERTIAN_DASAR_DAN_SIMBOL_FLOWCHART)

Dari gambar *flowchart* di atas terlihat bahwa suatu *flowchart* harus terdapat proses persiapan dan proses akhir. Dan yang menjadi topik dalam pembahasan ini adalah tahap proses. Karena kegiatan ini banyak mengandung variasi sesuai dengan kompleksitas masalah yang akan dipecahkan. Walaupun tidak ada kaidah-kaidah yang baku dalam penyusunan *flowchart*, namun ada beberapa anjuran yaitu:

- a. Hindari pengulangan proses yang tidak perlu dan logika yang berbelit sehingga jalannya proses menjadi singkat.
- b. Penggambaran *flowchart* yang simetris dengan arah yang jelas.
- c. Sebuah *flowchart* diawali dari satu titik START dan diakhiri dengan END.

Terdapat tiga macam operator pada *flowchart*:

a. Operator Numerik

Tabel 2.3 Macam-Macam Operator Numerik

| | |
|------|------------------|
| + | Penjumlahan |
| - | Pengurangan |
| * | Perkalian |
| / | Pembagian |
| ^ | Pangkat |
| sqrt | Akar pangkat dua |

b. Operator Hubungan

Tabel 2.4 Macam-Macam Operator Hubungan

| | |
|----|-------------------------|
| = | Sama dengan |
| # | Tidak sama dengan |
| < | Lebih kecil |
| > | Lebih Besar |
| <= | Lebih kecil sama dengan |
| >= | Lebih besar sama dengan |

c. Operator Logika

Tabel 2.5 Macam-Macam Operator Logika

| | |
|-----|--------------|
| AND | Logika DAN |
| OR | Logika ATAU |
| NOT | Logika LAWAN |

2.3.6 *Hosting dan Domain*

(Abdiansyah 2018) *Hosting* (disebut juga *Web Hosting* / sewa *hosting*) adalah penyewaan tempat untuk menampung data-data yang diperlukan oleh sebuah *website* dan sehingga dapat diakses lewat internet. Data disini dapat berupa *file*, gambar, *email*, aplikasi/program/*script* dan *database*.

(Krisianto 2012) *Domain* adalah identitas untuk mengakses sebuah *website*. Dalam bahasa sehari-hari kita sering menyebut *domain* sebagai alamat *website*. Contoh domain: *ugm.ac.id*, *google.com*, *indonesia.travel*, *wikipedia.org*, dan lain-lain.

Pada sebuah *website*, nama *domain* dan *web hosting* adalah satu kesatuan. Apabila yang tersedia hanya *web hosting*, maka *website* tidak akan dapat diakses. Apabila yang tersedia hanya *domain*, maka tidak akan ada *website* yang akan ditampilkan, karena halaman *website* tersimpan dalam *web hosting*.

2.3.7 *Black Box Testing*

Menurut Ramdani (2018, 69), “Teknik pengujian *Black-box* hanya akan berfokus pada fungsionalitas sebuah sistem. Di mana dalam skenario pengujian, penguji akan mencoba semua fungsional sebuah sistem dengan kondisi yang telah didefinisikan di awal”.

Menurut Desai dan Srivastava (2016, 116), “*Black box testing refers to the functional testing. Black box means a device you cannot see through; in testing parlance, it means that the software has to be tested without knowing what is inside it. The functional testing is performed based on the requirements document, as it contains the expected functionality and features*”.

Pengujian *black box* mengacu pada pengujian fungsional. *Black box* berarti perangkat yang tidak dapat Anda lihat; dalam pengujian bahasa, itu berarti bahwa perangkat lunak harus diuji tanpa mengetahui apa yang ada di dalamnya. Pengujian fungsional didasarkan pada dokumen persyaratan, karena berisi fungsionalitas dan fitur yang diharapkan.

2.4 Tinjauan Studi

2.4.1 Penggunaan *Bot Telegram* sebagai *Announcement System* pada Intansi Pendidikan (Soeroso, et al. 2017)

| No | Data Jurnal | Keterangan |
|----|------------------------------|---|
| 1 | Judul | Penggunaan <i>Bot Telegram</i> sebagai <i>Announcement System</i> pada Intansi Pendidikan |
| 2 | Jurnal | Seminar MASTER 2017 PPNS |
| 3 | Volume dan Halaman | Hal 45-48 |
| 4 | Tanggal & Tahun | Desember 2017 |
| 5 | Penulis | Hariyanto Soeroso, Afif Zuhri Arfianto, Novi Eka Mayangsari, Muhammad Taali |
| 6 | Penerbit | Politeknik Negeri Madiun |
| 7 | Tujuan Penelitian | Membuat <i>Announcement System</i> menggunakan <i>Bot Telegram</i> untuk meningkatkan pelayanan publik pada institusi pendidikan. |
| 8 | Lokasi dan Subjek Penelitian | |
| 9 | Perancangan Sistem | a. Metode <i>long-polling</i>
b. Metode <i>webhook</i>
c. Bahasa pemrograman PHP |
| 10 | Hasil Penelitian | Hasil dari penelitian ini adalah:
a. Menggunakan metode <i>long-polling</i> dan <i>webhook</i> dalam membuat <i>Bot Telegram</i> . |

| | | |
|----|----------------------|--|
| | | b. Metode <i>webhook</i> menghasilkan waktu respon yang lebih cepat dibanding metode <i>long-polling</i> . |
| 11 | Kekuatan Penelitian | Kekuatan penelitian adalah:
Penggunaan <i>Bot Telegram</i> memberikan informasi yang dibutuhkan dengan respon yang sangat cepat yaitu berkisar 0,97 sampai 2,46 detik. |
| 12 | Kelemahan Penelitian | Kelemahan penelitian adalah:
<i>Bot Telegram</i> ini hanya merespon perintah-perintah yang telah ditentukan saja, sehingga diharapkan mampu menerapkan <i>Artificial Intellegence</i> agar lebih interaktif. |
| 13 | Kesimpulan | Kesimpulan:
Ada dua cara dalam pembuatan <i>Bot Telegram</i> yaitu menggunakan metode <i>long-polling</i> dan <i>webhook</i> .
Pengujian menggunakan metode <i>long-polling</i> memiliki waktu respon yang lebih tinggi daripada metode <i>webhook</i> . |

2.4.2 Pencarian Informasi Wisata Daerah Bali menggunakan Teknologi

Chatbot (Paliwahet, Sukarsa and Putra 2017)

| No | Data Jurnal | Keterangan |
|----|--------------------|---|
| 1 | Judul | Pencarian Informasi Wisata Daerah Bali menggunakan Teknologi <i>Chatbot</i> |
| 2 | Jurnal | LONTAR KOMPUTER |
| 3 | Volume dan Halaman | Vol. 8 No. 3, Hal 144-153 |
| 4 | Tanggal & Tahun | Desember 2017 |
| 5 | Penulis | I Nyoman Satria Paliwahet, I Made Sukarsa, I Ketut Gede Darma Putra |
| 6 | Penerbit | Universitas Udayana |

| | | |
|----|------------------------------|--|
| 7 | Tujuan Penelitian | Pemanfaatan teknologi <i>chatbot</i> dengan memanfaatkan fitur <i>Fulltext Search Boolean Mode</i> sebagai alternatif pencarian informasi tentang wisata di Pulau Bali |
| 8 | Lokasi dan Subjek Penelitian | Pulau Bali |
| 9 | Perancangan Sistem | a. <i>Database MySQL</i>
b. Metode <i>Fulltext Search Boolean Mode</i> |
| 10 | Hasil Penelitian | Hasil dari penelitian ini adalah:
a. Menggunakan metode <i>Fulltext Search Boolean Mode</i> dalam proses pencocokan pola pada <i>chatbot</i> pencarian informasi wisata daerah Bali.
b. Perhitungan metode <i>Fulltext Search Boolean Mode</i> |
| 11 | Kekuatan Penelitian | Kekuatan penelitian adalah:

Penggunaan metode <i>Fulltext Search Boolean Mode</i> memiliki nilai relevansi yang dapat dijadikan acuan data pola yang paling relevan dari keseluruhan pola yang ada. |
| 12 | Kelemahan Penelitian | Kelemahan penelitian adalah:

Terdapat jawaban yang tidak sesuai karena pola kalimatnya tidak terdaftar dan <i>chatbot</i> tetap mengeluarkan jawaban sesuai dengan pola yang berkaitan atau mengandung kata pada pola lainnya. |
| 13 | Kesimpulan | Kesimpulan:

a. Sistem <i>chatbot</i> dengan menerapkan <i>Fulltext Search Boolean Mode</i> dari <i>MySQL</i> dapat diterapkan dengan baik dan memiliki nilai relevansi yang dapat dijadikan acuan data pola yang paling relevan. |

| | | |
|--|--|--|
| | | b. Jumlah pola yang didaftarkan mempengaruhi hasil yang akan dikeluarkan sebagai respon. Semakin banyak pola yang didaftarkan maka akan semakin tinggi kemungkinan dalam menemukan pola yang paling relevan. |
|--|--|--|

2.4.3 Pemanfaatan *Sentence-Similarity Measurement* untuk Proses Pencarian

Pola pada *Chatbot* berbasis *Pattern-Matching* (Dewi dan Setiaji 2014)

| No | Data Jurnal | Keterangan |
|----|------------------------------|--|
| 1 | Judul | Pemanfaatan <i>Sentence-Similarity Measurement</i> untuk proses pencarian pola pada <i>Chatbot</i> Berbasis <i>Pattern-Matching</i> |
| 2 | Jurnal | Seminar Nasional Teknologi Informasi dan Multimedia 2014 |
| 3 | Volume dan Halaman | Vol. 2 No. 1, |
| 4 | Tanggal & Tahun | 8 Februari 2014 |
| 5 | Penulis | Ayu Mutiara Oktavia Dewi, Bayu Setiaji |
| 6 | Penerbit | STMIK AMIKOM Yogyakarta |
| 7 | Tujuan Penelitian | Membuat representasi pengetahuan <i>chatbot</i> dan metode pencarian polanya menggunakan <i>sentence-similarity measurement</i> yang terpaket dalam RDBMS. |
| 8 | Lokasi dan Subjek Penelitian | |
| 9 | Perancangan Sistem | a. RDBMS MySQL
b. <i>Sentence-similarity measurement</i> menggunakan <i>bigram</i> |
| 10 | Hasil Penelitian | Hasil dari penelitian ini adalah: |

| | | |
|----|----------------------|--|
| | | Menggunakan <i>sentence-similarity measurement</i> dalam proses pencocokan antara masukan dengan pola yang ada di <i>table</i> . |
| 11 | Kekuatan Penelitian | <p>Kekuatan penelitian adalah:</p> <p>Penggunaan <i>sentence-similarity measurement</i> mencari kemiripan antara masukan dengan pola yang tersedia dengan skor/nilai <i>similarity</i> antara 0 sampai 1. Masukan yang memiliki skor 0 dengan pola yang tersedia tidak akan menghasilkan respon/jawaban apapun.</p> |
| 12 | Kelemahan Penelitian | <p>Kelemahan penelitian adalah:</p> <p>Terdapat jawaban yang tidak sesuai dengan fakta yang sebenarnya namun berdasarkan skor <i>similarity</i> tertinggi proses pencocokan pola dikatakan berhasil karena jumlah <i>pattern-template</i> yang digunakan masih sedikit.</p> |
| 13 | Kesimpulan | <p>Kesimpulan:</p> <ol style="list-style-type: none"> a. Penggunaan <i>sentence-similarity measurement</i> dalam proses pencocokan pola di <i>chatbot</i> adalah dengan mencari skor tertinggi pada pola. b. Beberapa masukan berbeda yang mengandung arti sama dapat diwakili oleh satu pola dalam pengetahuan. c. Diperlukan keragaman fakta yang dimasukkan sebagai <i>pattern-template</i> untuk dapat memberikan hasil yang lebih natural. |

2.4.4 Rangkuman Model Penelitian

Tabel 2.6 Rangkuman Jurnal

| Peneliti | Nama Jurnal | Tahun | Institusi | Judul dan Metode yang Digunakan | Kesimpulan |
|---|--|-------|--------------------------------|---|---|
| Hariyanto
Soeroso, Afif
Zuhri Arfianto,
Novi Eka
Mayangsari,
Muhammad
Taali | Seminar
MASTER
2017 PPNS
ISSN: 2548-
1509
e-ISSN: 2548-
6527 | 2017 | Politeknik
Negeri
Madiun | Penggunaan <i>Bot Telegram</i> sebagai <i>Announcement System</i> pada Intansi Pendidikan | Ada dua cara dalam pembuatan <i>Bot Telegram</i> yaitu menggunakan metode <i>long-polling</i> dan <i>webhook</i> .
Penguujian menggunakan metode <i>long-polling</i> memiliki waktu respon yang lebih tinggi daripada metode <i>webhook</i> . |
| I Nyoman
Satria
Paliwahet, I
Made Sukarsa,
I Ketut Gede
Darma Putra | LONTAR
KOMPUTER
ISSN: 2088-
1541
e-ISSN: 2541-
5832 | 2017 | Universitas
Udayana | Pencarian Informasi Wisata Daerah Bali menggunakan Teknologi <i>Chatbot</i> | Sistem <i>chatbot</i> dengan menerapkan <i>Fulltext Search Boolean Mode</i> dari MySQL dapat diterapkan dengan baik dan memiliki nilai relevansi yang dapat dijadikan acuan data pola yang paling relevan.

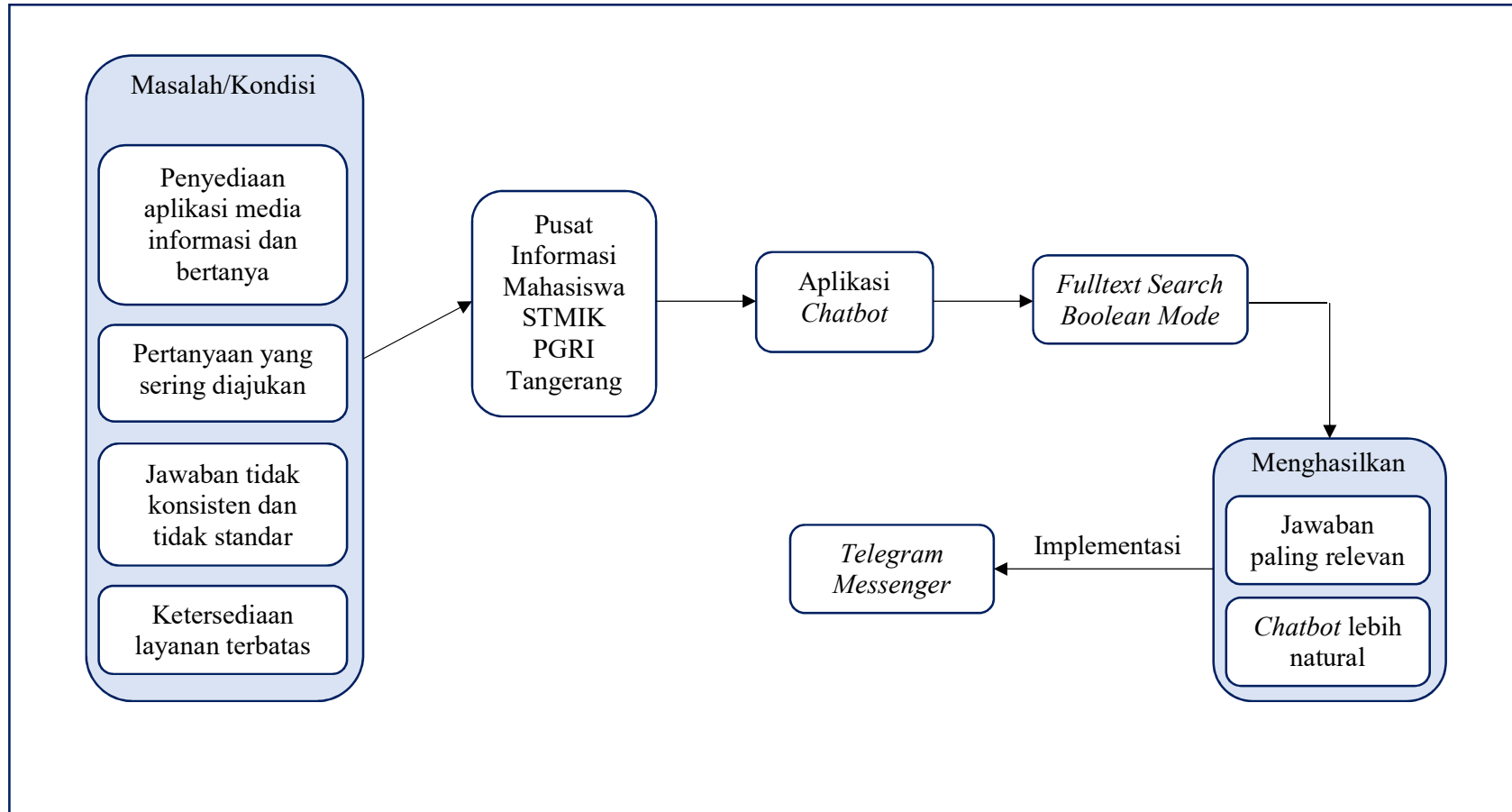
Jumlah pola yang didaftarkan mempengaruhi hasil yang akan dikeluarkan sebagai respon. Semakin banyak pola yang didaftarkan maka akan semakin tinggi |

| | | | | | |
|--|---------------------|------|-------------------------------|--|---|
| | | | | | kemungkinan dalam menemukan pola yang paling relevan. |
| Ayu Mutiara
Oktavia Dewi,
Bayu Setiaji | ISSN: 2302-
3805 | 2014 | STMIK
AMIKOM
Yogyakarta | Pemanfaatan
<i>Sentence-Similarity
Measurement</i> untuk
proses pencarian
pola pada <i>Chatbot</i>
Berbasis <i>Pattern-
Matching</i> | Penggunaan <i>sentence-similarity measurement</i> dalam proses pencocokan pola di <i>chatbot</i> adalah dengan mencari skor tertinggi pada pola.

Beberapa masukan berbeda yang mengandung arti sama dapat diwakili oleh satu pola dalam pengetahuan

Diperlukan keragaman fakta yang dimasukkan sebagai <i>pattern-template</i> untuk dapat memberikan hasil yang lebih natural. |

2.5 Kerangka Pemikiran



Gambar 2.4 Kerangka Pemikiran

Penulis mengidentifikasi beberapa masalah/kondisi saat ini antara lain penyediaan aplikasi sebagai media informasi dan bertanya, pertanyaan yang sering diajukan pengguna, jawaban yang diberikan karyawan satu dengan lainnya tidak standar dan tidak konsisten, serta jumlah karyawan dan ketersediaan layanan yang terbatas. Masalah-masalah tersebut secara garis besar dapat disimpulkan menjadi satu permasalahan umum yaitu tentang Pusat Informasi Mahasiswa di STMIK PGRI Tangerang. Dalam mengatasi permasalahan tersebut, penulis mengusulkan sistem aplikasi *chatbot* dengan menggunakan metode *Fulltext Search Boolean Mode*. Penggunaan metode tersebut dapat memberikan hasil respon/jawaban yang paling relevan dengan masukan dari pengguna serta membuat *chatbot* lebih natural, kemudian diimplementasikan ke aplikasi *chatbot* pada media *Telegram Messenger*.