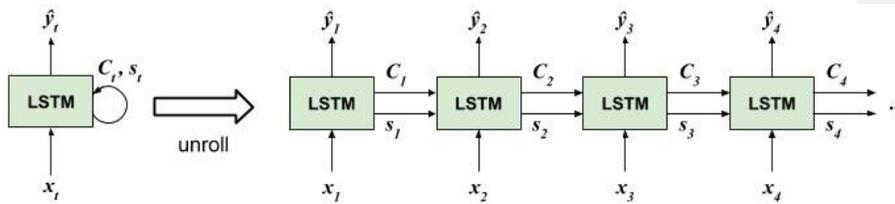


**BAB 3 PERANCANGAN DAN IMPLEMENTASI**

**3.1 Long Short Term Memory (LSTM)**

**LSTM** (*Long Short Term Memory*) adalah jenis modul pemrosesan lain untuk RNN. LSTM diciptakan oleh Hochreiter & Schmidhuber (1997) dan kemudian dikembangkan dan dipopulerkan oleh banyak periset. Seperti RNN, jaringan LSTM (*LSTM network*) juga terdiri dari modul-modul dengan pemrosesan berulang. Bedanya adalah modul-modul yang membentuk jaringan LSTM adalah modul LSTM:

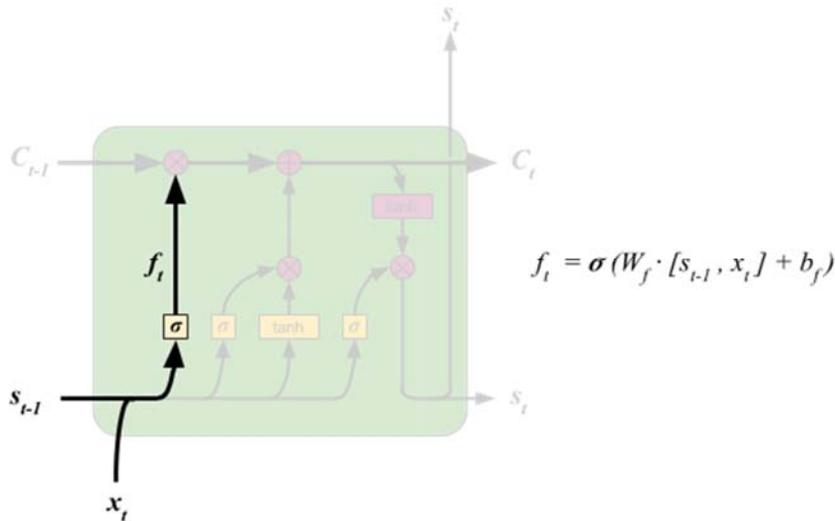


Gambar 3.1 Modul LSTM

Jaringan LSTM terdiri dari modul LSTM yang dipanggil secara berulang

Modul LSTM (satu kotak hijau) mempunyai pemrosesan yang berbeda dengan modul RNN biasa. Perbedaan lain adalah adanya tambahan sinyal yang diberikan dari satu langkah waktu ke langkah waktu berikutnya, yaitu konteks, direpresentasikan dengan simbol  $C_t$ .

Langkah pertama dalam LSTM adalah memutuskan informasi apa yang akan dibuang dari konteks  $C_{t-1}$ , dengan menggunakan gerbang sigmoid yang kita sebut “gerbang lupa” (*forget gate, f<sub>t</sub>*). Gerbang ini membaca nilai  $s_{t-1}$  dan  $x_t$ , dan menghasilkan angka antara 0 dan 1 untuk setiap elemen dalam  $C_{t-1}$ . Nilai 1 artinya “benar-benar jaga elemen ini” sementara 0 artinya “benar-benar singkirkan elemen ini.”



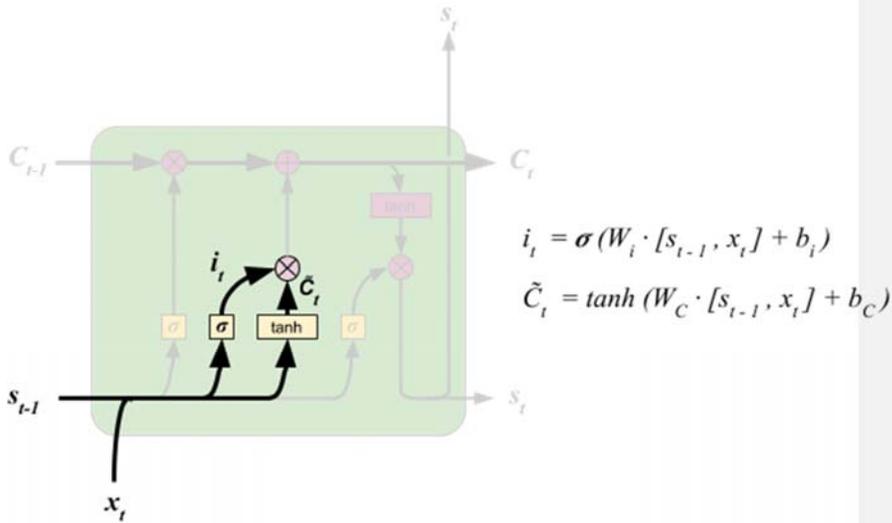
Gambar 3.2 Modul LSTM

Catatan: notasi  $[s_{t-1}, x_t]$  merupakan operasi konkatenasi; artinya kita menambahkan baris dari  $x_t$  dengan baris dari  $s_{t-1}$ .

Misalkan dalam contoh model bahasa yang mencoba memprediksi kata berikutnya berdasarkan semua kata sebelumnya:

Suatu elemen dalam  $C_{t-1}$  mungkin menyimpan informasi gender subjek saat ini, sehingga kata ganti yang benar dapat digunakan. Ketika kita melihat subjek baru, maka elemen lama dalam  $C_{t-1}$  bisa kita lupakan.

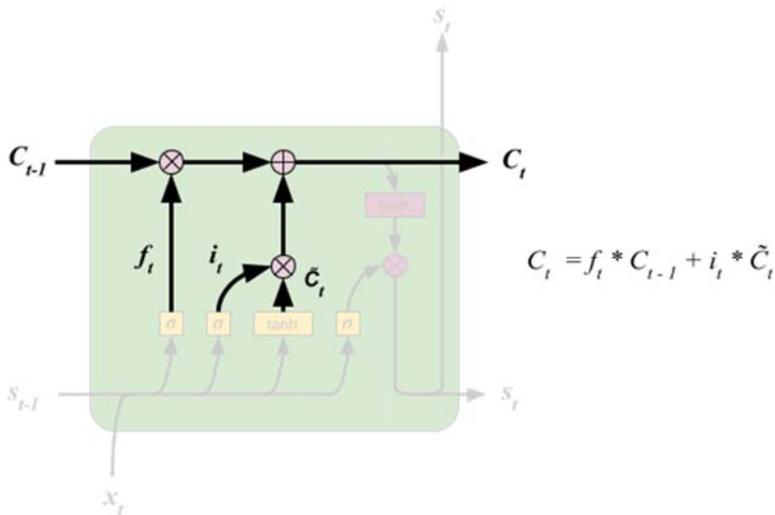
Langkah selanjutnya adalah memutuskan informasi baru apa yang akan kita gunakan di  $C_t$ . Proses ini memiliki dua bagian. Pertama, gerbang sigmoid yang disebut “gerbang masukan” (*input gate,  $i_t$* ) memutuskan nilai mana yang akan kita perbarui. Lalu sebuah lapis *tanh* menghasilkan kandidat vektor konteks baru,  $\tilde{C}_t$ , (dibaca: C tilde). Lalu kita gabungkan keduanya untuk membuat pembaruan ke konteks nanti.



Gambar 3.3 Modul LSTM

Dalam contoh model bahasa kita, kita menambahkan jenis kelamin subjek baru ke konteks, untuk menggantikan elemen yang lama yang kita lupakan.

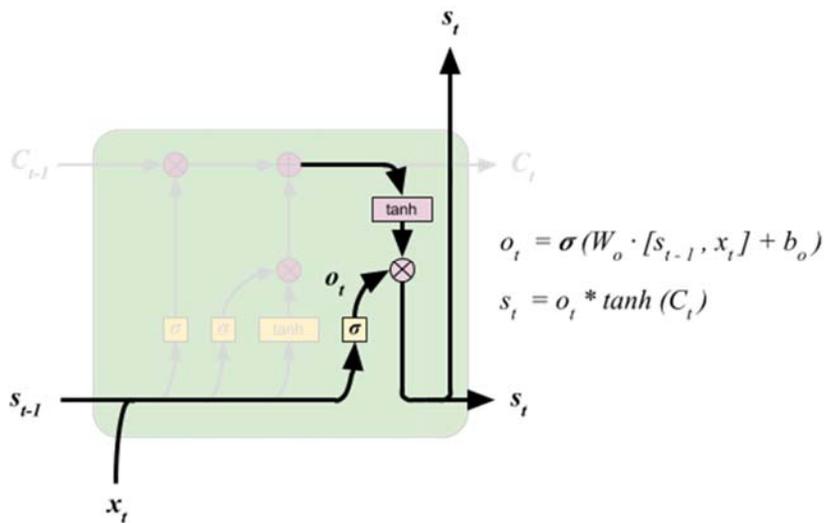
Sekarang saatnya untuk memperbarui konteks lama  $C_{t-1}$  ke konteks baru  $C_t$ . Kita kalikan konteks lama dengan  $f_t$  untuk melupakan hal-hal yang kita putuskan untuk dilupakan. Kita kalikan kandidat konteks baru kita  $\tilde{C}_t$  dengan  $i_t$  untuk memutuskan seberapa banyak kita akan menyertakan kandidat konteks baru. Lalu kita tambahkan keduanya.



Gambar 3.4 Modul LSTM

Dalam kasus model bahasa, ini adalah tempat kita benar-benar membuang informasi tentang gender subjek lama dan menambahkan informasi baru, seperti yang kita putuskan pada langkah sebelumnya.

Akhirnya, kita perlu memutuskan apa yang akan kita hasilkan. Output ini akan didasarkan pada nilai dalam konteks kita dan dilewatkan ke suatu filter. Pertama, kita jalankan gerbang sigmoid yang kita namakan gerbang output (*output gate,  $o_t$* ) untuk memutuskan bagian-bagian apa dari konteks yang akan kita hasilkan. Kemudian, kita lewatkan konteks melalui *tanh* untuk membuat nilainya menjadi antara -1 dan 1, dan kita kalikan dengan output gerbang sigmoid tadi sehingga kita hanya menghasilkan bagian yang kita putuskan.



Gambar 3.5 Modul LSTM

Untuk contoh model bahasa, ketika kita baru melihat subjek baru, mungkin ingin menyimpan informasi yang relevan untuk menghasilkan kata berikutnya, misalnya apakah subjeknya tunggal atau jamak, informasi gendernya, dsb.

### 3.1.1 Penggunaan LSTM pada Penelitian Ini

*Long short term memory network* (LSTM) adalah salah satu modifikasi dari *recurrent neural network* atau RNN. Banyak modifikasi dari RNN, tetapi LSTM merupakan salah satu yang populer di antaranya. LSTM hadir untuk melengkapi kekurangan RNN yang tidak dapat memprediksi kata berdasarkan informasi lampau yang disimpan dalam jangka waktu lama.

Dengan demikian, LSTM mampu mengingat kumpulan informasi yang telah disimpan dalam jangka waktu panjang, sekaligus menghapus informasi yang tidak lagi relevan. LSTM lebih efisien dalam memproses, memprediksi, sekaligus mengklasifikasikan data berdasarkan urutan waktu tertentu.

Perbedaan mendasar dari LSTM dan RNN adalah bahwa LSTM melengkapi kekurangan-kekurangan yang dimiliki oleh pendahulunya, *recurrent neural network*. Dengan kata lain, persoalan jangka waktu penyimpanan tidak menjadi permasalahan dalam LSTM.

*Long Short-Term Memory* (LSTM) merupakan model varian dari *Recurrent Neural Network* (RNN). LSTM muncul karena dapat mengingat informasi jangka panjang (long term dependency) LSTM menggantikan simpul *hidden layer* di RNN dengan sel LSTM yang berfungsi untuk menyimpan informasi sebelumnya. Dalam LSTM terdapat tiga gerbang yang mengendalikan penggunaan dan memperbarui informasi terdahulu yaitu *input gate*, *forget gate*, dan *output gate*. Sel memori dan tiga gerbang dirancang untuk dapat membaca, menyimpan, dan memperbarui informasi terdahulu.

Arsitektur LSTM dikembangkan sebagai solusi dari masalah *vanishing gradient* yang ditemui pada RNN konvensional. *Vanishing gradient* disebabkan karena gradien yang semakin mengecil hingga *layer* terakhir membuat nilai bobot tidak berubah sehingga menyebabkan tidak pernah memperoleh hasil yang lebih baik atau konvergen. Sebaliknya gradien yang semakin membesar menyebabkan nilai bobot pada beberapa *layer* juga ikut membesar sehingga algoritma optimasi menjadi divergen atau disebut *exploding gradient*

### 3.1.2 *Recurrent Neural Network* (RNN)

Perbedaan mendasar dari LSTM dan RNN adalah bahwa LSTM melengkapi kekurangan-kekurangan yang dimiliki oleh pendahulunya, *recurrent neural network*, yang tidak dapat memprediksi data berdasarkan informasi yang telah disimpan dalam waktu cukup lama. Dengan kata lain, persoalan jangka waktu penyimpanan tidak menjadi permasalahan dalam LSTM.

Sistem yang menerapkan LSTM dapat memproses, memprediksi, dan mengklasifikasikan informasi berdasarkan data deret waktu. Sesuai dengan konsepnya, LSTM dapat mengingat dan menghapus data-data lawas yang sudah tidak relevan lagi. Dengan demikian, manajemen informasi akan lebih komplet sekaligus aktual.

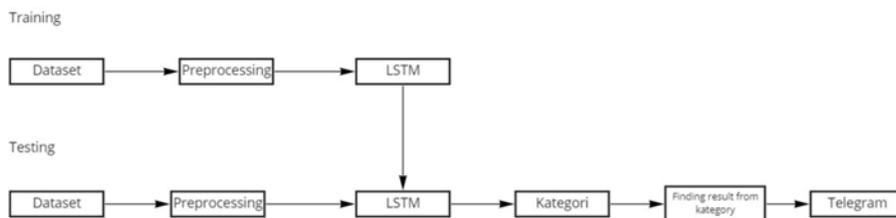
Perbedaan berikutnya dari LSTM dan RNN terletak pada lapisan rantainya. Jika RNN memungkinkan satu neuron untuk memproses satu input data pada satu output data, LSTM tidak berlaku demikian. LSTM memiliki berbagai gerbang yang dapat menambah kumpulan informasi dan menggabungkannya.

Ada empat gerbang dalam sistem LSTM, yakni *forget gate*, *input gate*, *input modulation gate*, serta *output gate*. Keempat gerbang tersebut mempunyai fungsi dan tugasnya masing-masing dalam mengumpulkan, mengklasifikasi, dan memproses data. Tak hanya mempunyai empat gerbang tersebut, LSTM juga memiliki *internal cell state* yang berfungsi untuk menyimpan informasi pilihan dari unit sebelumnya

### 3.2 Arsitektur Sistem

Seperti yang telah dijelaskan pada bab sebelumnya, sistem *Chatbot* yang dirancang pada penelitian ini akan digunakan sebagai media pemasaran ITHB. Diharapkan sistem ini dapat memudahkan calon mahasiswa dan staf pemasaran ITHB. Gambaran dari arsitektur sistem dapat dilihat pada Gambar 3.1.

Proses pelatihan dimulai dari pemuatan *file dataset training* untuk melakukan *preprocessing*. Hasil dari *preprocessing* akan menjadi masukan untuk *LSTM* yang dibuat, selanjutnya setiap neuron masukan diberi nilai *weight* dan *bias* awalan. Nilai masukan akan melewati proses perhitungan pada *neural network* untuk menghasilkan sebuah model *LSTM* dengan nilai *weight* dan *bias* yang sudah ditetapkan. Model LSTM yang sudah dilatih kemudian diuji menggunakan *dataset testing*. *Dataset testing* sebelumnya melewati tahap *preprocessing* agar dapat diolah model *LSTM*. Hasil pengujian kemudian akan dicatat untuk melakukan menampilkan kategori dan kategori tersebut sistem akan mencari jawaban dari kategori yang ditampilkan, kemudian di kirim ke *Telegram*.

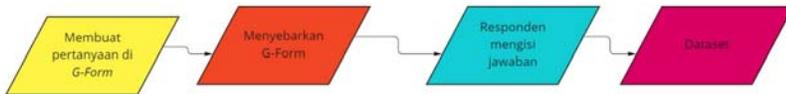


Gambar 3.6 Arsitektur Sistem

#### 3.2.1 Mekanisme pengumpulan Dataset

Untuk mengumpulkan dataset, dibuat pertanyaan di *Google Form*, mengenai pertanyaan apa saja yang biasanya calon mahasiswa tanyakan saat akan melanjutkan pendidikan ke perguruan tinggi. Kemudian disebar *Google Form* yang akan diisi oleh responden melalui berbagai sosial media. Lalu responden mengisi pertanyaan yang biasanya

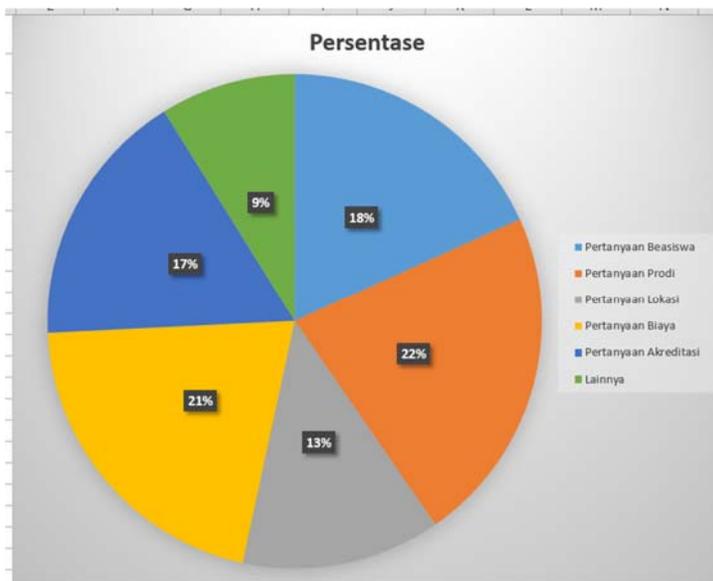
ditanyakan oleh calon mahasiswa sebelum mendaftar kuliah. Setelah terkumpul jawaban dari responden maka itu yang dijadikan dataset.



Gambar 3.7 Mekanisme Pengumpulan Dataset

### 3.2.2 Kategori dalam sistem Chatbot

Setelah dataset terkumpul, diberi label kategori untuk setiap pertanyaan. Ada 5 kategori yang di pilih yaitu: beasiswa, biaya, lokasi, akreditasi, dan program studi. Alasan diambilnya 5 kategori dari banyak pertanyaan yang masuk dikarenakan persentase pertanyaan di 5 kategori tersebut adalah yang paling sering ditanyakan. Di gambar 3.3 adalah persentase dalam diagram lingkaran yang dibuat menggunakan *tools Excel*.



Gambar 3.8 Persentase kategori pertanyaan

### 3.2.3 Preprocessing Dataset Yang Digunakan

Teks yang ada di dalam *dataset* harus diubah terlebih dahulu menjadi nilai yang dapat diolah oleh *LSTM*. Proses *preprocessing* dilakukan sebelumnya untuk mengubah pola kalimat berbentuk teks menjadi suatu matriks deretan angka. Tahap pertama yang dilakukan adalah *tokenizing* atau pembagian kalimat menjadi kata-kata penyusunnya. Selanjutnya penghapusan

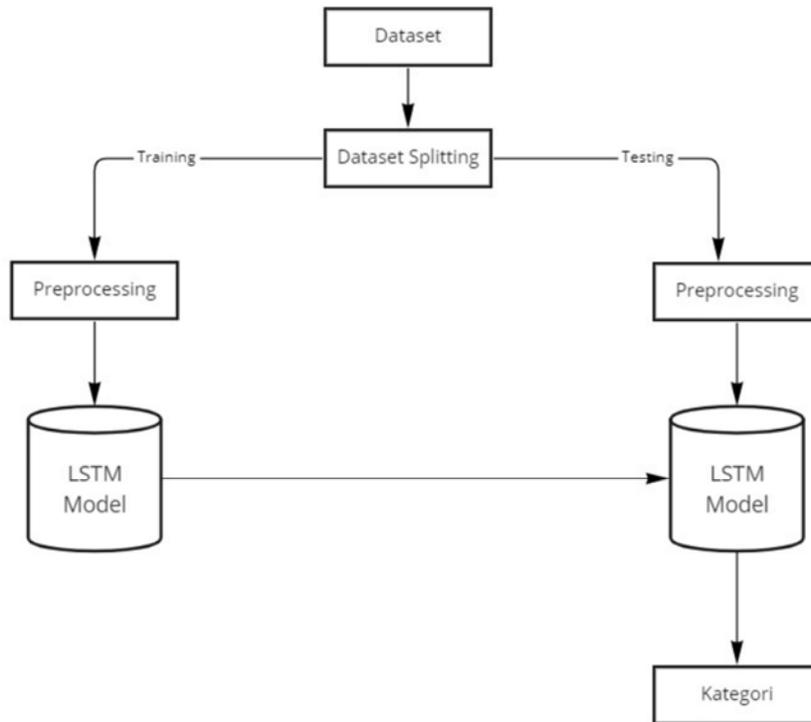
tanda baca. Lalu kata-kata tersebut dilakukan *Case Folding* untuk mengubah huruf besar menjadi huruf kecil agar keseluruhan kata tetap seragam. Selanjutnya *Stemming* atau pengubahan kata menjadi kata dasar untuk mempertegas keseragaman data. Proses *preprocessing* dapat dilihat pada Tabel 3.1.

Tabel 3.1 Contoh *Preprocessing*

Proses	Contoh
Sentence	Di mana ITHB berada?
Tokenizing	["Di", "mana", "ITHB", "Berada?"]
Penghapusan Tanda Baca	["Di", "mana", "ITHB", "Berada"]
Case Folding	["di", "mana", "ithb", "berada"]
Stemming	["di", "mana", "ithb", "ada"]

#### 3.2.4 Pelatihan Model LSTM

Dalam melatih model *LSTM* membutuhkan dataset sebagai acuan bagi model dalam menentukan jenis pertanyaan. Dataset terdiri dari lima kategori yaitu biaya, beasiswa, lokasi, akreditasi, dan program studi. Hasil dari *preprocessing* akan menjadi masukan untuk *Long Short-Term Memory* yang dibuat, selanjutnya setiap neuron masukan diberi nilai *weight* dan *bias* awalan. Nilai masukan akan melewati proses perhitungan pada *neural network* untuk menghasilkan sebuah model LSTM dengan nilai *weight* dan *bias* yang sudah ditetapkan. Model *Long Short Term Memory* yang sudah dilatih kemudian diuji menggunakan *dataset testing*. *Dataset testing* sebelumnya melewati tahap *preprocessing* agar dapat diolah model *Long Short Term Memory*. Hasil pengujian kemudian akan dicatat untuk melakukan perhitungan penilaian model atau *scoring*. Kemudian *output* nya berupa kategori. diagram arsitektur dari sistem dapat dilihat pada Gambar 3.2.

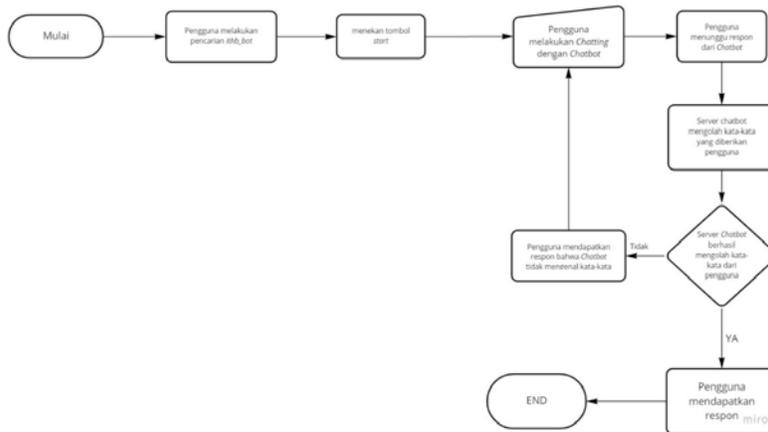


Gambar 3.9 *Training dan Testing LSTM*

### 3.2.5 Proses Interaksi pada *Chatbot*

Sistem akan bekerja setelah pengguna mengirimkan pesan kepada *Chatbot* melalui *Telegram* sebagai aplikasi yang digunakan untuk berkomunikasi dengan pengguna. Pengguna dapat mengunduh *Telegram* melalui App Store ataupun Play Store, lalu melakukan pencarian akun *ithb\_bot* di pencarian aplikasi *Telegram*. Melalui aplikasi *Telegram* pengguna dapat mengirimkan pesan pertanyaan terkait ITHB. Alur kerja sistem dapat dilihat pada Gambar 3.3. Pengguna melakukan pencarian akun *Telegram ithb\_bot*. Setelah menemukan akun *ithb\_bot* pengguna akan diarahkan ke dinding obrolan dimana pengguna harus klik tombol *start* untuk mulai bertanya. Setelah menekan tombol *start* yang ada pada layar obrolan, calon mahasiswa dapat mulai memasukkan pertanyaan, pengguna akan menunggu jawaban dari *chatbot*, pertanyaan yg masuk akan di teruskan ke server untuk mengolah jenis pertanyaan yang masuk, kemudian server berhasil mengolah pertanyaan dari calon mahasiswa jika iya maka calon mahasiswa akan menerima jawaban melalui telegram, jika server tidak berhasil

mengerti jenis pertanyaan dari calon mahasiswa maka akan tampil jawaban bahwa *chatbot* tidak mengerti dari pertanyaan pengguna.



Gambar 3.10 Interaksi Pada *Chatbot*

### 3.2.6 Proses Yang Terjadi pada *Server*

Proses yang terjadi dalam *server* diawali dengan dataset sebagai acuan bagi model dalam menentukan jenis pertanyaan. Dataset disini terdiri dari lima label yaitu biaya, beasiswa, lokasi, akreditasi, dan program studi. Hasil dari *preprocessing* akan menjadi masukan untuk *Long Short-Term Memory* yang dibuat. Hasil dari tahap *LSTM* adalah berupa kategori yaitu biaya, beasiswa, lokasi, dan program studi. Keluaran kategori tersebut diproses lagi melalui tahap *finding result from category* di tahap ini kategori dengan akurasi lebih besar dari 70%. Kategori tersebut yang akan menentukan jawaban dari pertanyaan. Setelah mendapatkan jawaban, jawaban itu yang akan dikirim ke *Telegram*. Proses yang terjadi pada server dapat dilihat pada Gambar 3.6.



Gambar 3.11 Proses yang Terjadi Pada *Server*

### 3.3 Integrasi *LSTM* ke *Telegram*

Hasil dari modeling *LSTM* kemudian di export ke file pickl, dari file pickl tersebut kemudian dipanggil pada aplikasi chatbot yang sudah dibuat menggunakan python. Setiap chat yang masuk, maka akan dinilai oleh model yang sudah disimpan pada file pickle, dan

dari sana akan mengeluarkan output label apa dan nilai accuracy nya. Kemudian Chatbot akan mengirimkan pesan sesuai dengan label yang dihasilkan dari LSTM model tersebut.

#### Listing program Integrasi LSTM ke *Telegram*

```
1. pickle.dump(model, open('/content/drive/MyDrive/LSTM CHTBOT/model.pkl', 'wb'))
2. myModel = pickle.load(open('/content/drive/MyDrive/LSTM CHTBOT/model.pkl', 'rb'))
3. #!/usr/bin/env python
4. # -*- coding: utf-8 -*-
5.
6. import telebot
7. import pickle
8.
9. from telebot import types
10.
11. TOKEN_BOT = '5380213544:AAE3K6tH10YOK1DjuAKtb3Jq03sPoJ73Ewo' #ithb_bot
12. bot = telebot.TeleBot(TOKEN_BOT)
13.
14.
15. labels = ['jurusan', 'beasiswa', 'biaya', 'lokasi', 'akreditasi']
16.
17. myModel = pickle.load(open('/content/drive/MyDrive/LSTM CHTBOT/model.pkl', 'rb'))
18.
19. def bot_pkl(text):
20.     new_complaint = [text]
21.     seq = tokenizer.texts_to_sequences(new_complaint)
22.     padded = pad_sequences(seq, maxlen=MAX_SEQUENCE_LENGTH)
23.     pred = myModel.predict(padded)
24.     return (labels[np.argmax(pred)])
25.
26. def bot_check(text):
27.     new_complaint = [text]
28.     seq = tokenizer.texts_to_sequences(new_complaint)
29.     padded = pad_sequences(seq, maxlen=MAX_SEQUENCE_LENGTH)
30.     pred = myModel.predict(padded)
31.     return (labels[np.argmax(pred)], np.max(pred))
32.
33. @bot.message_handler(commands=['help'])
34. def action_help(message):
35.     bot.send_message(message.chat.id, "* Selamat Datang di Bot Kampus ITHB\n",
36. parse_mode="MARKDOWN")
37.
38. @bot.message_handler(commands=['start'])
39. def action_help(message):
40.     bot.send_message(message.chat.id, "*Menu *START*, Selamat Datang di Bot Kampus ITHB\n",
41. parse_mode="MARKDOWN")
42.
43. def kirim_beasiswa(chatid):
44.     msg_beasiswa = f'''
45. *BEASISWA HARAPAN BANGSA*
46.
47. Institut Teknologi Harapan Bangsa selalu mendukung setiap mahasiswanya melalui beasiswa
48. dengan memperhatikan prestasi yang dicapai. Berbeda dengan kampus lain, ITHB memberikan
49. beasiswa tidak hanya di awal masuk kuliah (JPA, Beasiswa UTBK), tetapi juga selama studi
50. sampai lulus (GGB).
51.
52. Beasiswa-beasiswa tersebut diberikan untuk mendukung mahasiswa berprestasi agar dapat
53. menempuh pendidikan sarjana demi mencapai karier global di jalur cepat namun memiliki
54. keterbatasan finansial. Berikut ini adalah beasiswa-beasiswa yang diberikan oleh Institut
55. Teknologi Harapan Bangsa.
56.
57. *JPA (Jalur Prestasi Akademik)*
58. Beasiswa potongan sampai dengan 100% biaya pengembangan untuk siswa berprestasi (ranking 1-
59. 10) selama kelas X & XI.
```

### BAB 3 PERANCANGAN DAN IMPLEMENTASI

```
51.
52. *JPO Basket (Jalur Prestasi Olahraga Basket)*
53. Beasiswa 50-100% selama 8 semester untuk siswa dengan prestasi olah raga minimal di tingkat provinsi.
54.
55. *BPRG (Beasiswa Putra/I Rohaniwan & Guru)*
56. Beasiswa untuk putra/i rohaniwan dan guru, potongan 50% selama 8 semester (syarat dan ketentuan berlaku).
57.
58. *BSS (Beasiswa Subsidi Silang)*
59. Beasiswa potongan 50% biaya pengembangan bagi siswa yang merupakan saudara kandung dari mahasiswa aktif ITHB. Dan beasiswa potongan 25% bagi siswa yang merupakan saudara kandung dari alumnus ITHB.
60.
61. *GGB (Gerakan Gereja Belajar)*
62. Beasiswa penuh 100% selama 8 semester, tersedia untuk 50 orang/tahun akademik, bagi siswa yang lolos seleksi, dengan kriteria berprestasi secara akademik, aktif dalam pelayanan dan organisasi.
63.
64. *Bantuan Finansial*
65. Selain beasiswa-beasiswa di atas, ITHB juga menawarkan beberapa jenis bantuan finansial seperti: Pinjaman Mahasiswa, Beasiswa Atlet, Beasiswa Putra-putri Rohaniwan dan Guru, dan lain-lain
66.     '''
67.     bot.send_message(chatid, msg_bea, parse_mode="MARKDOWN")
68.
69. def kirim_jurusan(chatid):
70.     msg_jurusan = f'''
71.     1. Teknik Informatika
72.     2. Sistem Informasi
73.     3. Teknik Komputer
74.     4. Teknik Industri
75.     5. Manajemen Rantai Pasok
76.     6. Desain Komunikasi Visual
77.     7. Fashion Apparel Desain
78.     8. Akuntansi
79.     9. Online Bussines
80.     10. Manajemen
81.     '''
82.     bot.send_message(chatid, msg_jurusan, parse_mode="MARKDOWN")
83.
84. def kirim_lokasi(chatid):
85.     msg_loc = f'''*ITHB*\nJl. Dipati Ukur No.80, Dago, Kecamatan Coblong, Kota Bandung, Jawa Barat 40132
86.     '''
87.     bot.send_message(chatid, msg_loc, parse_mode="MARKDOWN")
88.     bot.send_location(chatid, -6.889349 , 107.616099)
89.
90. def kirim_biaya(chatid):
91.     path = ('/content/drive/MyDrive/LSTM CHTBOT/biaya.jpg')
92.     # path = "biaya.jpg"
93.     photo = open(path, "rb")
94.     bot.send_photo(chatid, photo)
95.
96. def kirim_akreditasi(chatid):
97.     bot.send_chat_action(chatid, "upload_photo")
98.     # path = "akreditasi_1.jpg"
99.     # path2 = "akreditasi_2.jpg"
100.     path = ('/content/drive/MyDrive/LSTM CHTBOT/akreditasi_1.jpg')
101.     path2 = ('/content/drive/MyDrive/LSTM CHTBOT/akreditasi_2.JPG')
102.     photo = open(path, "rb")
103.     photo2 = open(path2, "rb")
104.     bot.send_photo(chatid, photo)
105.     bot.send_photo(chatid, photo2)
106.
107. @bot.message_handler(func=lambda message:True)
108. def handle_message(message):
109.     text = message.text.lower().split()
110.     chat_id = message.chat.id
```

```
111.     list_basiswa = ['basiswa']
112.     list_jurusan = ['jurusan', 'jurusannya']
113.     list_loc = ['tempat', 'lokasi', 'lokasinya']
114.     list_biaya = ['duit', 'biaya', 'bea', 'harga']
115.     list_akreditasi = ['akreditasi']
116.
117.     try:
118.         if any(elem in text for elem in list_basiswa):
119.             msg = f"Berikut merupakan info tentang *Basiswa*"
120.             bot.send_message(message.chat.id, msg, parse_mode="MARKDOWN")
121.             kirim_basiswa(message.chat.id)
122.
123.         elif any(elem in text for elem in list_jurusan):
124.             msg = f"Berikut merupakan info tentang *Jurusan*"
125.             bot.send_message(message.chat.id, msg, parse_mode="MARKDOWN")
126.             kirim_jurusan(message.chat.id)
127.
128.         elif any(elem in text for elem in list_loc):
129.             msg = f"Berikut merupakan info tentang *Lokasi*"
130.             bot.send_message(message.chat.id, msg, parse_mode="MARKDOWN")
131.             kirim_lokasi(message.chat.id)
132.
133.         elif any(elem in text for elem in list_biaya):
134.             msg = f"Berikut merupakan info tentang *Biaya*"
135.             bot.send_message(message.chat.id, msg, parse_mode="MARKDOWN")
136.             kirim_biaya(message.chat.id)
137.
138.         elif any(elem in text for elem in list_akreditasi):
139.             msg = f"Berikut merupakan info tentang *Akreditasi*"
140.             bot.send_message(message.chat.id, msg, parse_mode="MARKDOWN")
141.             kirim_akreditasi(message.chat.id)
142.
143.     else:
144.         cek = bot_check(message.text.lower())
145.         if cek[1]>0.7:
146.             msg = f"Berikut merupakan info tentang *{cek[0]}*"
147.             bot.send_message(message.chat.id, msg, parse_mode="MARKDOWN")
148.             if cek[0] == 'basiswa':
149.                 kirim_basiswa(message.chat.id)
150.             elif cek[0] == 'jurusan':
151.                 kirim_jurusan(message.chat.id)
152.             elif cek[0] == 'lokasi':
153.                 kirim_lokasi(message.chat.id)
154.             elif cek[0] == 'biaya':
155.                 kirim_biaya(message.chat.id)
156.             elif cek[0] == 'akreditasi':
157.                 kirim_akreditasi(message.chat.id)
158.         else:
159.             # msg = f"nilai accuracy hanya *{cek[1]}*"
160.             msg = f"Maaf, kami kurang mengerti dengan yang ada maksud.\nApakah
ada yang bisa kami bantu?"
161.             bot.send_message(message.chat.id, msg, parse_mode="MARKDOWN")
162.
163.     except:
164.         msg = f"Mohon maaf terjadi kesalahan"
165.         bot.send_message(message.chat.id, msg, parse_mode="MARKDOWN")
166.
167.
168.
169.     print('bot start running')
170.
171.     def main_loop():
172.         bot.polling(True)
173.         while True:
174.             time.sleep(1)
175.
176.     if __name__ == '__main__':
177.         try:
178.             main_loop()
```

```

179.         except Exception as err:
180.             print ('Internet Error')
    
```

### 3.4 Data Pengguna Telegram

Berdasarkan DataIndonesia.id Telegram berada dalam urutan ke-3 pengguna aplikasi pesan terbanyak. Meski popularitasnya masih kalah dibanding WhatsApp, Telegram memiliki sejumlah keunggulan seperti mampu mengirim data berukuran besar, mampu membuat grup komunikasi daring dengan kapasitas 200 ribu peserta, sinkronisasi cepat, dan sebagainya. Aplikasi ini juga memiliki fitur ruang obrolan rahasia (*Secret Chat*), di mana pesan yang dikirim akan terenkripsi secara otomatis sehingga tidak ada pihak ketiga yang bisa membaca isi pesan tanpa persetujuan pengguna. Berdasarkan data lansiran *Business of Apps*, pengguna Telegram di seluruh dunia sudah mencapai 500 juta orang pada 2021. Jumlah ini meningkat 25% dibandingkan tahun sebelumnya yang masih berjumlah 400 juta orang [12].



Gambar 3.12 Pengguna Telegram

Berikut ini adalah kelebihan dari aplikasi *Telegram* :

1. Terdapat Fitur *Chatbot*

Fitur unik yang ada di Telegram salah satunya adalah chat bot. Chat bot ini dapat berfungsi untuk mendapatkan informasi mengenai hal yang diperlukan, seperti cuaca, menerjemahkan bahasa, berita terkini, dan yang lainnya. Pemilik akun juga dapat dihubungkan dengan orang lain yang mempunyai minat yang sama melalui chat bot ini. [10]

## 2. Memiliki Layanan *Cloud*

Telegram mempunyai penyimpanan sendiri dengan kapasitas 1,5 GB. Artinya pengguna dapat menyimpan file seperti teks, gambar, video, dokumen dan yang lainnya. Dengan fitur cloud tersebut, user dapat dengan bebas berganti akun tanpa perlu takut file yang tersimpan akan hilang. [10]

## 3. Memiliki Fitur *Chat* Rahasia

Pemilik akun dan pengguna lain dapat melakukan obrolan rahasia dengan fitur secret chat. Layanan ini memungkinkan untuk berkomunikasi lebih privat, tanpa disimpan oleh Cloud. Fitur ini tidak dapat diretas oleh hacker, dan pesan akan otomatis terhapus dalam jangka waktu tertentu. [10]

### 3.5 Chatbot Penting Bagi Bisnis

Chatbot memudahkan operasional bisnis dan menghemat biaya customer service. Teknologi ini sangat memungkinkan untuk menjawab berbagai jenis pertanyaan ataupun masalah yang dihadapi pelanggan.

Ada beberapa keuntungan *Chatbot*, antara lain :

#### 1. Tersedia 24 jam

Dikutip dari *Entrepreneur.com* sebanyak 68% konsumen akan pergi ke bisnis kompetitor, apabila tidak bisa dilayani dengan cekatan. Untuk itulah, bot diciptakan agar selalu melayani konsumen tanpa batas waktu tertentu.

#### 2. Menghemat Biaya

*Chatbit* sangat efektif bagi bisnis yang ingin menghemat biaya. Hal ini didukung hasil riset milik *Forrest Consulting* yang menyatakan bila sebuah perusahaan yang menyediakan layanan *Chatbot* dalam proses bisnis, dapat menekan biaya *Customer Service* sebesar 30% [5].

#### 3. Meningkatkan Jumlah Penjualan

Teknologi bot tidak hanya meningkatkan kepuasan pelanggan terhadap layanan bisnis, tetapi juga berpengaruh pada reputasi. Dengan adanya reputasi bisnis yang baik, maka kesempatan untuk menambah penjualan lebih besar.

Dikutip dari hasil riset yang sama yaitu *Forrester Consulting*, menyatakan bahwa selain dapat menghemat biaya *Customer Service*, layanan bot juga berpotensi untuk memperoleh hasil investasi sebesar 305% [5].

Alasannya dari efektivitas yang dihasilkan dari teknologi bot, yang kemudian menghasilkan peningkatan pada *Customer Experience*.

Menurut *Entrepreneur.com* salah satu alasan pelanggan pergi adalah pelayanan *customer service* yang buruk. Gambar 3.12 adalah data *why they leave* [5]



Gambar 3.13 Data *why they leave*[5]

### 3.6 Implementasi Sistem

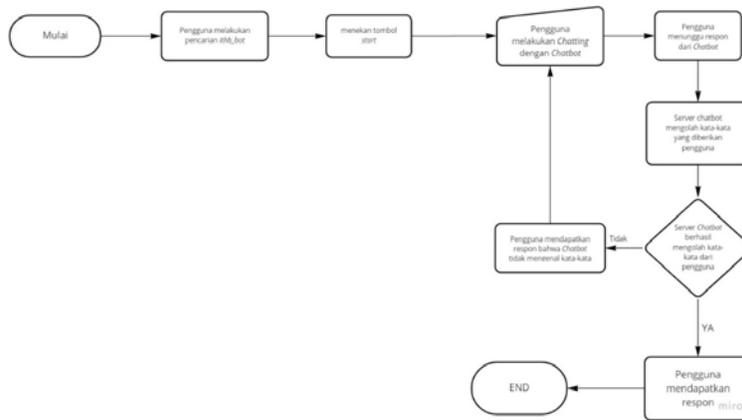
Pada bagian perancangan dan implementasi sistem, akan dijelaskan mengenai perancangan cara kerja sistem, dan perancangan antarmuka yang dapat dilihat pada Gambar 3.5. Rancangan tersebut kemudian akan diimplementasikan sehingga menghasilkan sistem *Chatbot digital marketing* yang terintegrasi dengan antarmuka berbasis *Telegram*.

Secara umum suatu *chatbot* bekerja dengan cara memindai kata kunci dalam pesan pengguna, lalu *chatbot* memberi balasan dengan kata kunci yang paling cocok atau pola kata yang paling mirip dari data yang sudah disiapkan. Implementasi dari tampilan awal *chatbot* dapat dilihat pada Gambar 3.6 dan untuk implemetasi tampilan proses tanya jawab dapat dilihat Gambar 3.7.

*Chatbot* yang dikembangkan dalam penelitian ini dirancang untuk dapat bekerja dengan cara sebagai berikut:

1. Pengguna melakukan pencarian akun *ithb\_bot* di *Telegram*.
2. Setelah menemukan akun *ithb\_bot*, kemudian klik tombol *start* untuk memulai tanya jawab.
3. Setelah menekan tombol *start*, pengguna dapat bertanya mengenai *ITHB*, dengan mengetikkan pertanyaan-pertanyaan.
4. Setelah mengetikkan pertanyaan, pengguna menunggu jawaban dari *chatbot*.
5. Server *chatbot* mengolah pertanyaan yang ditanyakan oleh pengguna.
6. Jika pengguna tidak mendapatkan jawabannya, *Chatbot* akan memberikan respon tidak mengenali pertanyaan tersebut.
7. Namun jika *chatbot* dapat mengenali pertanyaan, maka pengguna akan mendapatkan respon sesuai pertanyaannya.

BAB 3 PERANCANGAN DAN IMPLEMENTASI



Gambar 3.14 Flowchart Implementasi Chatbot



Gambar 3.15 Tampilan Awal Chatbot

