E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 07, Nomor 03, September 2022 : 671–679

IMPLEMENTASI SISTEM BROADCAST MESSAGE MENGGUNAKAN PYTHON DAN REDIS PUB/SUB

Jovan Millenno Claudiyap[1)], Pratyaksa Ocsa Nugraha Saian[2)]
[1, 2)]Fakultas Teknologi Informasi,Universitas Kristen Satya Wacana,
Jawa Tengah, Indonesia
e-mail: 672018086@student.uksw.edu[1)],pratyaksa.ocsa@uksw.edu [2)]

ABSTRAK

Semakin berkembangnya perusahaan ritel maka semakin banyak juga toko-toko yang tersebar di berbagai daerah. Hal ini mengharuskan penyampaian informasi dari kantor pusat ke toko harus dapat tersampaikan dengan baik dan merata. Menyikapi hal tersebut, maka pada penelitian ini dilakukan pembangunan sistem broadcast message yang bertujuan untuk mengirimkan pesan secara realtime ke komputer toko. Sistem ini dibangun menggunakan Redis server yang diimplementasikan pada arsitektur publish-subscribe dengan metode SDLC Waterfall. Hasil dari penelitian ini yaitu sebuah sistem yang dapat mengirimkan pesan dan menampilkan secara realtime pada komputer toko berupa pop up, sehingga pesan dapat langsung tersampaikan. Sistem ini telah diuji menggunakan metode Blackbox dan User Acceptance Test (UAT) dengan total indeks rata-rata 79% yang menunjukkan bahwa sistem broadcast message dapat dikatakan layak dan menjawab permasalahan yang ada.

***Kata Kunci**: Broadcast Message, Pub/Sub, Python, Redis.*

ABSTRACT

The more developed retail companies are, the more shops are scattered in various regions. This requires the delivery of information from the head office to the store must be conveyed properly and evenly. In response to this, in this study, a broadcast message system was developed which aims to send messages in real time to the store computer. This system is built using a Redis server which is implemented on a publish-subscribe architecture with the SDLC Waterfall method. The results of this study are a system can send messages and display in real time on the store computer in the form of pop ups, so that messages can be delivered directly. This system has been tested using the Blackbox and User Acceptance Test (UAT) method with an average total index of 79% which indicates that the broadcast message system can be said to be feasible and answers the existing problems

***Keywords**:Broadcast Message, Pub/Sub, Python, Redis.*

I. PENDAHULUAN

Pesatnya perkembangan zaman membuat dunia bisnis ikut berkembang dengan cepat. Banyak perusahaan di bidang ritel bersaing untuk meningkatkan fasilitas dan pelayanan kepada *customer* [1]. Pada toko ritel *modern*, terdapat fasilitas perangkat komputer untuk menunjang operasional toko, seperti menjalankan aplikasi kasir dan mengolah data [2].

Semakin berkembangnya perusahaan ritel maka semakin banyak juga toko-toko yang tersebar di berbagai daerah. Sehingga dalam penyampaian informasi dari kantor pusat ke toko harus dapat tersampaikan dengan baik dan merata. Secara umum media yang sering digunakan untuk penyampaian informasi yaitu E-mail maupun *Social Media*. Namun alternatif tersebut tidak efektif jika digunakan dalam kasus ini, karena ada kemungkinan pesan tidak dibaca langsung oleh karyawan toko.

Untuk menyelesaikan permasalahan yang dihadapi, maka pada penelitian ini akan dilakukan pembangunan sistem *broadcast message* yang bertujuan untuk mengirimkan pesan secara *realtime* ke komputer toko. Sistem ini akan dibangun menggunakan Redis yang diimplementasikan pada arsitektur *publish-subscribe* dengan memanfaatkan fitur *message broker*. Fitur ini berfungsi menjadi perantara pengiriman pesan dari kantor pusat ke toko melalui *channel* tertentu.

Redis merupakan teknologi penyimpanan struktur data yang cukup populer karena melakukan pemrosesan data dalam memori sehingga memiliki kecepatan yang baik [3]. Selain itu Redis juga bersifat *open-source* sehingga dapat meminimalisir biaya.

Pembangunan sistem ini juga memanfaatkan *micro-framework* Flask dan *library* Redis pada Python, karena penggunaan sintaks yang sederhana dan adanya kumpulan kode yang dapat langsung digunakan untuk mempercepat proses pembangunan sistem [4].

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 07, Nomor 03, September 2022 : 671–679

Penelitian yang berjudul Implementasi Arsitektur *Publish Subscribe* Pada *Constrained Application Protocol* (COAP) di Lingkungan *Internet of Things* (IoT), Fauzi dan Adhitya melakukan penerapan arsitektur Pub/Sub pada CoAP. Hal ini diterapkan untuk mencegah aktivitas yang dapat mengganggu kinerja node server dikarenakan banyaknya *client* yang melakukan *request* data sehingga *resource* digunakan secara bersamaan. Dengan penerapan PubSub dapat mengurangi nilai *latency* dan efisiensi overhead yang lebih baik dibandingkan dengan CoAP konvensional yang masih menggunakan *request-respond* [5].

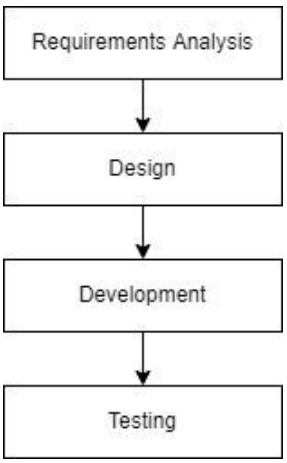
Penelitian yang berjudul Penggunaan Python Web Framework Flask Untuk Pemula, Rahadian menjelaskan bagaimana instalasi, fitur dan kelebihan serta kekurangan dalam penggunaan *micro-framework* Flask Python untuk pengembangan aplikasi berbasis web. Flask merupakan *framework* yang mudah dipelajari dan dapat dijalankan tanpa harus memiliki komputer dengan spesifikasi yang tinggi. Hal ini dapat mengurangi biaya serta mempercepat pengembangan aplikasi [6].

Penelitian yang berjudul Desain *Model View Controller* Dalam Implementasi Redis Server oleh Mulki, Ari dan Arief menjelaskan bagaimana implementasi Redis server dengan pendekatan Model View Controller (MVC) menggantikan penggunaan database relasional. Database relasional memiliki kelemahan dalam kecepatan akses data, karena masih disimpan di dalam harddisk. Namun berbeda dengan Redis yang memproses data dalam memori sehingga memiliki kecepatan yang lebih baik [7].

Pada beberapa penelitian sebelumnya, Redis Server digunakan sebagai media penyimpanan data menggantikan database *relational* dan arsitektur Pub/Sub diterapkan untuk efisiensi dalam distribusi data ke server. Pada penelitian yang akan dilakukan penulis, fungsi Pub/Sub pada Redis server akan dimanfaatkan pada pengembangan sistem *Broadcast Message* sebagai perantara untuk mempercepat distribusi data pesan ke banyak penerima (toko cabang) sekaligus. Selain itu penulis juga memanfaatkan *micro-framework* Python, yaitu Flask untuk mempercepat pembuatan *web service.*

II. METODE PENELITIAN

Pada penelitian ini menggunakan metode *Software Development Life Cycle* (SDLC) *Waterfall* [10]. SDLC *Waterfall* dipilih karena pada pengembangan sistem, *requirements* sudah ditentukan di awal dan tidak ada penambahan pada saat *development.* Tahapan dapat dilihat pada gambar 1:



Gambar 1 Tahapan metode penelitian SDLC Waterfall

Tahapan rancangan Sistem *Broadcast Massage* menggunakan SDLC *Waterfall* dapat diuraikan sebagai berikut:

- Tahapan pertama yaitu *Requiremets Analysis*. Pada tahap ini penulis melakukan analisa dan mengumpulkan informasi terkait kebutuhan sistem untuk menyelesaikan permasalahan yang ada. Pengumpulan informasi dengan cara observasi melalui beberapa dokumen seperti jurnal dan halaman web dari sumber terpercaya serta melakukan diskusi dengan beberapa praktisi IT.

- Tahap kedua yaitu Design. Setelah terkumpul informasi yang valid, maka pada tahap ini penulis membuat rancangan desain untuk *software architecture* dengan membuat alur sistem seperti gambaran bagan arsitektur,

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 07, Nomor 03, September 2022 : 671–679

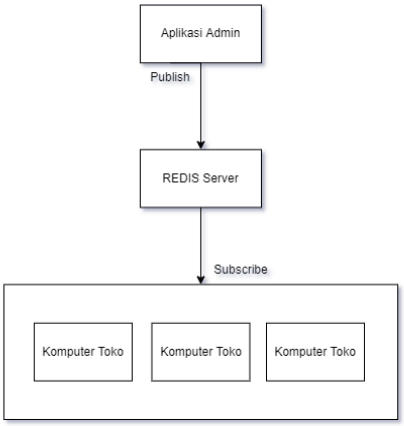
flowchart dan *activity* diagram serta pembuatan desain *user interface* agar memudahkan dalam pembuatan sistem di tahap selanjutnya.

- Tahap ketiga yaitu *Development*, pada tahap ini rancangan yang ada pada tahap kedua akan diimplementasikan penulis pada pembuatan sistem. Pembuatan sistem dimulai dari pembuatan *frontend* yang akan dibangun memanfaatkan HTML, CSS dan Javascript. Setelah bagian *frontend* selesai, akan dilanjutkan dengan pembuatan pada sisi *backend*. Pada bagian ini penulis membuat koneksi menggunakan Python untuk menghubungkan aplikasi utama dan aplikasi *client* dengan Redis Server.

- Tahap keempat yaitu Testing. Sistem yang telah selesai dibuat akan diuji menggunakan metode *Black Box*, dengan cara melakukan pengetesan pada pengiriman dan penerima pesan. Hal ini untuk memastikan sistem yang dibangun sudah sesuai rancangan. Selain itu penulis juga akan mengumpulkan data uji melalui kuesioner kepada user, untuk mendapatkan respons pendapat user secara langsung terhadap penggunaan aplikasi [11].

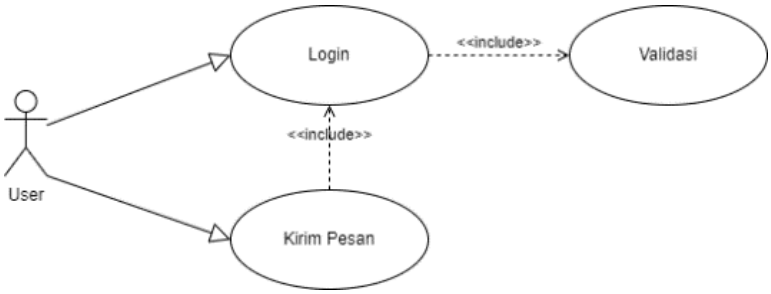
III. HASIL DAN PEMBAHASAN

Pada penelitian ini, menghasilkan sistem *Broadcast Message* yang dijalankan pada jaringan intranet atau jaringan internal perusahaan. Pembangunan aplikasi ini menggunakan teknologi dari bahasa pemrograman Python dengan *framework* Flask untuk membangun *web service* dan Redis server sebagai *message broker* dengan memanfaatkan fitur Pub/Sub.



Gambar 2 Arsitektur Sistem *Broadcast Message*

Gambar 2 merupakan arsitektur dari sistem *broadcast message* yang akan diimplementasikan. Terdapat 2 program yang dikembangkan yaitu pada sisi admin yang digunakan pada kantor pusat untuk mengirimkan pesan (*publish*) dan program pada sisi toko cabang untuk menerima dan menampilkan pesan (*subscribe*). Redis server berfungsi untuk menerima dan meneruskan pesan pada channel yang sudah tujuan.

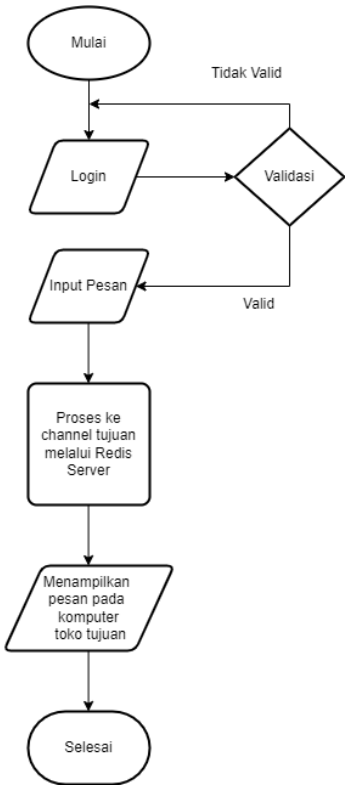


Gambar 3 *Use Case* Sistem *Broadcast Message*

Gambar 3 merepresentasikan interaksi antara *user* dan sistem yang dibangun, serta memberikan gambaran mengenai fungsi yang terdapat pada sistem. Pada sistem ini terdapat dua fungsi yaitu login dan kirim pesan.

E-ISSN : 2540 - 8984

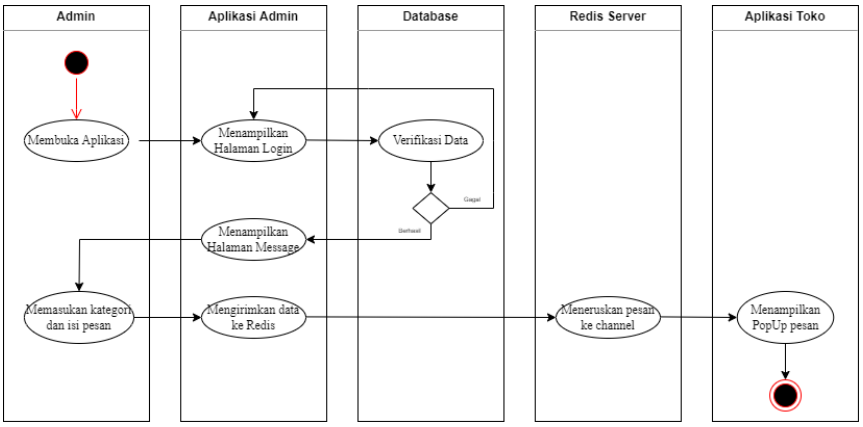
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 07, Nomor 03, September 2022 : 671–679



Gambar 4 *Flowchart* Sistem *Broadcast Message*

Gambar 4 memperlihatkan *flowchart* dari alur sistem. Dimulai dengan admin melakukan login pada halaman login, jika berhasil admin dapat memilih kategori dan isi pesan pada halaman *message*. Kemudian pesan akan diteruskan oleh Redis server ke aplikasi toko cabang untuk ditampilkan.

Untuk gambaran lebih jelas mengenai sistem yang dibangun dapat dilihat pada Gambar 5 berikut ini:



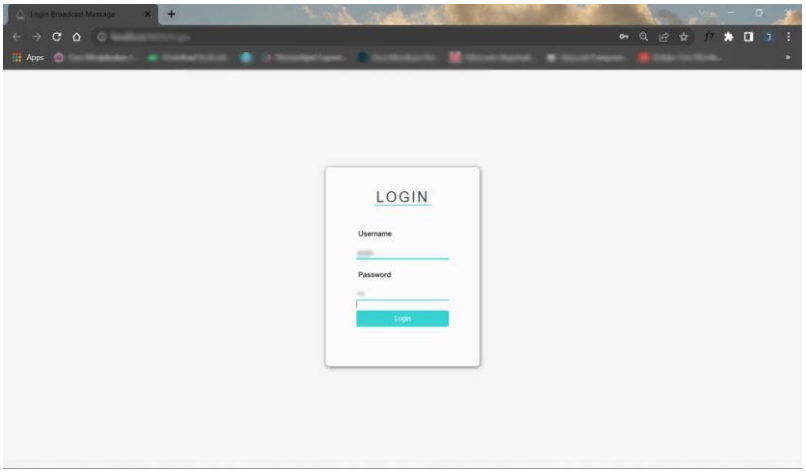
Gambar 5 *Activity Diagram* Sistem *Broadcast Message*

Gambar 5 memperlihatkan tahapan proses pada arsitektur sistem sebelum dan sesudah pesan terkirim. Pada saat aplikasi dijalankan, sistem akan menampilkan halaman login. Setelah admin berhasil login, sistem akan menampilkan halaman *message*. Pada halaman ini admin dapat memasukkan pesan dan kategori. Selanjutnya sistem akan

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 07, Nomor 03, September 2022 : 671–679

meneruskan data pada Redis server yang akan dikirimkan ke *channel* tujuan. Pesan yang terkirim akan ditampilkan secara *realtime* melalui *pop up message box* pada komputer toko cabang.



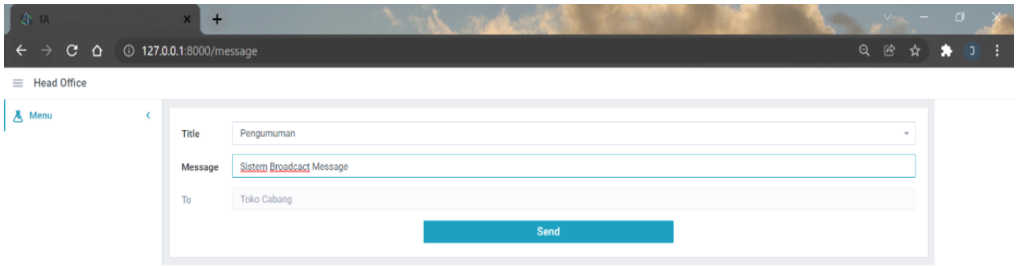
Gambar 6 Tampilan Halaman *Login*

Gambar 6 menampilkan halaman login yang menjadi halaman awal ketika aplikasi dijalankan. Pada halaman login, admin diharuskan mengisi *username* dan *password* yang menjadi syarat untuk dapat mengakses halaman selanjutnya.

Kode Program 1. Kode program *Backend* pada sisi Admin

```
1  import redis
2  conn = redis.Redis(host='192.168.102.70', port=6379,db=0)
3  def sendMessage(to,title,message):
4   data=[]
5   data.append(title)
6   data.append(message)
7   try:
8      conn.publish(to,str(data))
9      return jsonify({'message': 'Kirim Pesan Berhasil'})
10  except:
11     return jsonify({'message': 'Kirim Pesan Gagal'}),500
```

Kode Program 1 merupakan potongan kode program dari *backend web service.* Dapat dilihat pada baris 1 terdapat import redis yang digunakan untuk memanfaatkan fungsi pada *library* Redis, seperti fungsi untuk membuat koneksi, *publish* dan *subscribe.* Baris 2 digunakan untuk membuat koneksi agar sistem terhubung dengan Redis server. Dalam membuat koneksi diperlukan parameter seperti host, port dan nomor indeks database. Baris 8 merupakan kode untuk *publish message* dengan parameter berupa nama *channel* dan isi dari *message.*



Gambar 7 Tampilan Aplikasi *Broadcast Message* Admin

Gambar 7 merupakan tampilan *front end* dari aplikasi *broadcast message* pada sisi admin yang dibuat dengan sederhana sehingga lebih mudah untuk digunakan. Terdapat beberapa kategori yang tersedia seperti pengumuman, info harga dan info diskon. Selain itu terdapat juga *text field* untuk isi pesan.

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 07, Nomor 03, September 2022 : 671–679

Kode Program 2. Kode program *Backend* pada sisi Toko

```
1  import redis
2  import ast
3  import pyautogui
4  conn = redis.Redis(host='192.168.102.70', port=6379,db=0)
5  p = conn.pubsub()
6  p.subscribe('Toko')
7  while True:
8        message = p.get_message()
9        if message and not message['data'] == 1:
10           data = message['data'].decode('utf-8')
11           value = ast.literal_eval(data)
12           pyautogui.alert(value[1], value[0])  # always returns "OK"
```

Kode Program 2 merupakan potongan kode *backend* pada aplikasi toko yang berfungsi untuk menampilkan *pop up* berisi pesan informasi. Pada aplikasi toko terdapat beberapa penggunaan *library* tambahan seperti ast dan pyautogui. Pyautogui digunakan untuk menampilkan pesan informasi dalam bentuk *pop up message box*. Pada baris 5 dan baris 6 digunakan untuk inisialisasi nama *channel*.

Message yang dikirimkan melalui Redis server akan diterima dalam tipe data *byte*. Maka dari itu pada baris 10, *message* yang masuk akan diubah ke dalam bentuk *string* agar lebih mudah dikelola.

Gambar 8 Tampilan Aplikasi *Broadcast Message* Toko

Gambar 8 merupakan tampilan dari pop up message box. Message box otomatis muncul ketika message diterima dari Redis server.

Pengujian dengan metode Blackbox dilakukan pada sistem ini untuk memastikan fungsionalitas utama dari sistem berjalan dengan baik.

Hasil pengujian dapat dilihat pada Tabel 1.

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 07, Nomor 03, September 2022 : 671–679

TABEL I
HASIL PENGUJIAN *BLACKBOX*

ID	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Status Pengujian
X1	Mengisi kategori dan isi pesan lalu menekan tombol "Send"	Sistem menerima dan menampilkan notifikasi "Pesan berhasil terkirim" dan menampilkan popup isi pesan pada sisi toko	Sistem menampilkan notifikasi "Pesan berhasil terkirim" dan menampilkan popup isi pesan pada sisi toko	Valid
X2	Mematikan Redis server kemudian mengisi kategori dan isi pesan lalu menekan tombol "Send"	Sistem menerima dan menampilkan notifikasi "Pesan gagal terkirim, silahkan periksa koneksi"	Sistem memproses dan menampilkan notifikasi "Pesan gagal terkirim, silahkan periksa koneksi"	Valid
X3	Mengisi *username* dan *password* yang benar pada halaman login, lalu menekan tombol "Login"	Sistem memvalidasi *username* dan *password* sesuai yang terdaftar. Jika sesuai akan menampilkan halaman kirim m*essage*.	Sistem berhasil memvalidasi *username* dan *password* lalu menampilkan halaman kirim *message*.	Valid
X4	Mengisi *username* dan *password* yang salah pada halaman login lalu menekan tombol "Login"	Sistem memvalidasi *username* dan *password* sesuai yang terdaftar. Jika tidak sesuai maka sistem akan memunculkan *popup* yang bertuliskan "Username dan Password anda tidak sesuai".	Sistem berhasil memvalidasi *username* dan *password* yang salah kemudian menampilkan *popup* yang bertuliskan "Username dan Password anda tidak sesuai".	Valid

Untuk mengetahui apakah implementasi sistem *Broadcast Message* dapat meningkatkan atensi karyawan toko terhadap pesan informasi yang dikirimkan oleh admin kantor pusat, maka penulis melakukan pengujian menggunakan metode *User Acceptance Test* (UAT). Metode ini bertujuan untuk mendapatkan jawaban user terhadap sistem yang dibangun[12]. Metode ini dilakukan dengan cara memberikan kuesioner atau angket skala Likert yang digunakan untuk memberikan beberapa pertanyaan kepada user. Untuk lebih jelasnya dapat dilihat pada Tabel 2.

TABEL II
HASIL RESPONS

No	Pertanyaan	Jawaban				
		SS	**S**	**C**	**TS**	**STS**
1.	Apakah menurut anda, adanya aplikasi Broadcast Message ini dapat mempermudah admin kantor pusat untuk mengirimkan pesan informasi ke banyak Toko cabang sekaligus?	8	9	3	-	-
2.	Apakah menurut anda, adanya aplikasi Broadcast Message ini cocok digunakan oleh perusahaan untuk mengirimkan pesan informasi ke Toko cabang, menggantikan peran Email maupun Social Media?	4	9	6	1	-
3.	Apakah menurut anda, dengan mengirimkan pesan informasi melalui aplikasi Broadcast Message bisa mendapatkan perhatian karyawan Toko cabang secara langsung?	4	8	7	1	-

Setelah pengambilan data dari hasil kuesioner, maka akan dilanjutkan dengan analisis interval dengan memberikan skor pada jawaban yang ada sebagai berikut :

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 07, Nomor 03, September 2022 : 671–679

1. Sangat Setuju (S) : 5 skor
2. Setuju (S) : 4 skor
3. Cukup (C) : 3 skor
4. Tidak Setuju (TS) : 2 skor
5. Sangat Tidak Setuju (STS) : 1 skor

Dari hasil data yang diperoleh, maka akan diolah dengan perhitungan jumlah responden dikalikan dengan total skor sesuai jawaban responden. Hasil Perhitungan dapat dilihat pada Tabel 3.

TABEL III
PERHITUNGAN SKALA LIKERT

Pertanyaan ke-	Nilai Skor					Total Skor	Nilai Indeks ((Total Skor / Skor Maksimum) x 100) %
	SS (5 Skor)	S (4 Skor)	C (3 Skor)	TS (2 Skor)	STS (1 Skor)		
1.	40	36	9	-	-	85	85%
2.	20	36	18	2	-	76	76%
3.	20	32	21	2	-	75	75%

Interval Penilaian [12]:

- 0% - 19,99% = Sangat Tidak Setuju
- 20% - 39,99% = Tidak Setuju
- 40% - 59,9% = Kurang Setuju
- 60% - 79,9 = Setuju
- 80% - 100% = Sangat Setuju

Dari hasil perhitungan pada Tabel 3 maka setiap pertanyaan memiliki total indeks rata-rata 79% atau dapat diartikan sesuai interval penilaian yaitu “Setuju”. Sehingga dapat disimpulkan bahwa implementasi Sistem *Broadcast Message* mendapatkan respons yang baik dari *user*.

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, fitur Pub/Sub pada Redis difungsikan sebagai *message broker* untuk mengantar pesan dari admin ke toko cabang. Pengujian *Blackbox* yang valid serta melalui metode *User Acceptance Test* (UAT) dengan indeks rata-rata 79% maka dapat disimpulkan bahwa sistem *broadcast message* dapat dikatakan layak dan menjawab permasalahan yang ada.

Diharapkan dalam pengembangan maupun penelitian berikutnya, ada beberapa saran yang dapat dipertimbangkan seperti: 1) Menambahkan fitur keamanan pada sistem, 2) Pengembangan aplikasi pada platform lain seperti mobile, 3) Penambahan fitur riwayat pesan, 4) Penambahan fitur *feedback* pada sistem untuk mengetahui *client* yang sudah menerima pesan.

DAFTAR PUSTAKA

[1] J. L. Putra, M. Raharjo, T. A. A. Sandi, R. Ridwan, and R. Prasetyo, “Implementasi Algoritma Apriori Terhadap Data Penjualan Pada Perusahaan Retail,” *J. Pilar Nusa Mandiri*, vol. 15, no. 1, pp. 85–90, 2019, doi: 10.33480/pilar.v15i1.113.

[2] M. Abdurahman, “Sistem Informasi Pengolahan Data Pembelian Dan Penjualan Pada Toko Koloncucu Ternate,” *IJIS - Indones. J. Inf. Syst.*, vol. 2, no. 1, 2017, doi: 10.36549/ijis.v2i1.22.

[3] F. Febriyani, E. S. Pramukantoro, and F. A. Bachtiar, “Perbandingan Kinerja Redis, Mosquitto, dan MongoDB sebagai Message Broker pada IoT Middleware | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,” *J-Ptiik.Ub.Ac.Id*, vol. 03, no. 07, pp. 6816–6823, 2019.

[4] D. F. Ningtyas and N. Setiyawati, “Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request Flask Framework Implementation in Development Purchasing Approval Request Application,” *J. Janitra Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra.v1i1.120.

[5] M. Fauzi and A. Bhawiyuga, "Implementasi Arsitektur Publish Subscribe Pada Constrained Application Protocol (COAP) di Lingkungan Internet of Things (IoT)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 7, pp. 7060–7067, 2019.
[6] R. Irsyad, "Penggunaan Python Web Framework Flask Untuk Pemula," 2018, doi: 10.31219/osf.io/t7u5r.
[7] M. I. Zulfa, A. Fadli, A. W. Wardhana, T. Elektro, and U. J. Soedirman, "Desain Model View Controller Dalam Implementasi Redis," vol. 4, no. November, pp. 11–20, 2019.
[8] R. S. F. Jannatin, A. Suharsono, and A. Bhawiyuga, "Implementasi Publish-Subscribe Pada Delay Tolerant Network (DTN)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 2, pp. 118–124, 2017, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/51/27.
[9] B. P. Putra and Y. A. Susetyo, "Implementasi Api Master Store Menggunakan Flask, Rest Dan Orm Di Pt Xyz," *Sistemasi*, vol. 9, no. 3, p. 543, 2020, doi: 10.32520/stmsi.v9i3.899.
[10] Y. Firmansyah and U. Udi, "Penerapan Metode SDLC Waterfall Dalam Pembuatan Sistem Informasi Akademik Berbasis Web Studi Kasus Pondok Pesantren Al-Habib Sholeh Kabupaten Kubu Raya, Kalimantan Barat," *J. Teknol. dan Manaj. Inform.*, vol. 4, no. 1, 2017, doi: 10.26905/jtmi.v4i1.1605.
[11] W. S. Dharmawan, D. Purwaningtias, and D. Risdiansyah, "Penerapan Metode SDLC Waterfall Dalam Perancangan Sistem Informasi Administrasi Keuangan Berbasis Desktop," *J. Khatulistiwa Inform.*, vol. 6, no. 2, pp. 159–167, 2018, doi: 10.31294/khatulistiwa.v6i2.160.
[12] B. Priyatna, A. Lia Hananto, M. Nova, P. Studi Sistem Informasi, and U. Buana Perjuangan Karawang, "Application of UAT (User Acceptance Test) Evaluation Model in Minggon E-Meeting Software Development," *Systematics*, vol. 2, no. 3, pp. 110–117, 20