

ABSTRAK

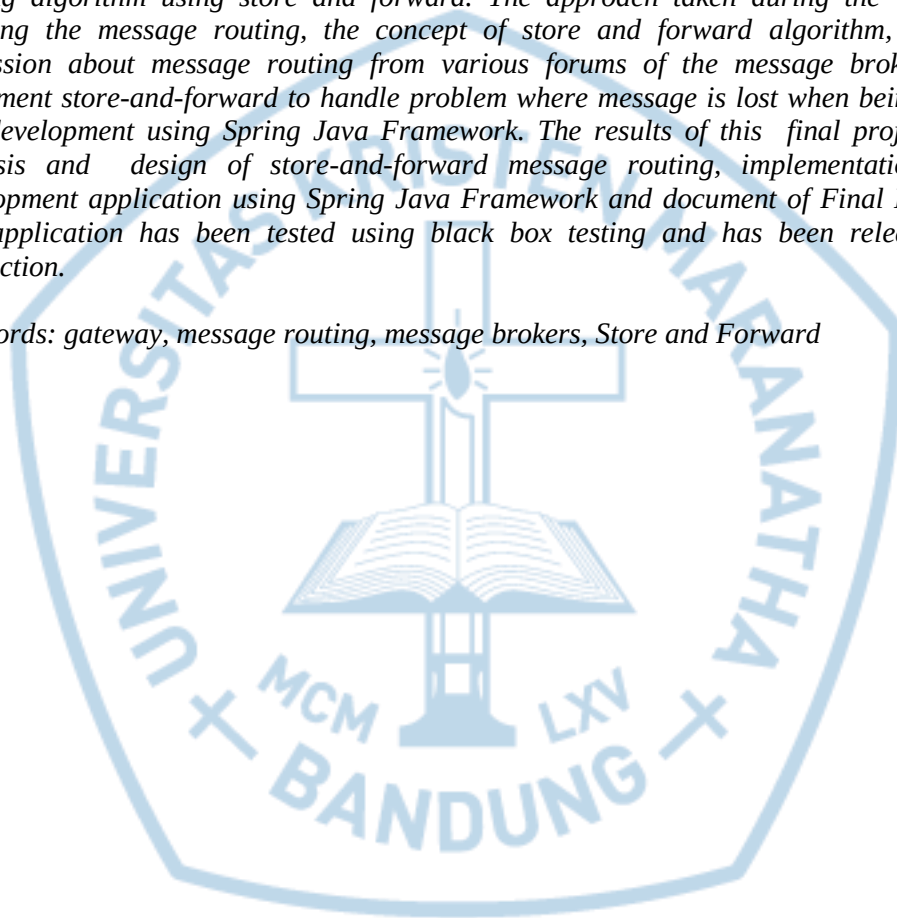
Store and forward adalah salah satu algoritma *messaging*, dimana pengiriman pesan oleh pengirim pesan(*sender*) disimpan terlebih dahulu sebelum diteruskan ke penerima pesan(*receiver*). Algoritma *store and forward* ini dapat mengatasi kendala pengiriman pesan seperti tidak tersedianya destinasi atau destinasi dalam keadaan tidak aktif. *Store and forward* mengatasi pesan yang hilang saat proses pengiriman berlangsung, pesan yang hilang berarti ada pesan yang dikirim oleh pengirim namun tidak sampai pada penerima pesan. Pesan yang dikirim akan melalui *gateway* dan *gateway* akan memastikan destinasi pesan aktif dan meneruskan pesan tersebut ke destinasi. Pengerjaan Tugas Akhir ini adalah menganalisis, mendesain dan mengimplementasikan *message routing* menggunakan algoritma *store and forward*. Pendekatan yang dilakukan selama mengerjakannya adalah mempelajari *message routing*, konsep *store and forward*, konsep implementasi menggunakan *framework* Java yaitu *Spring*, *research* dan diskusi *message routing* dari berbagai forum mengenai *message broker* dan mengimplementasikan *store-and-forward* untuk mengatasi kendala pesan hilang pada saat pengiriman pesan. Hasil dari pelaksanaan Tugas Akhir adalah implementasi hasil analisis, desain *store and forward message routing* dan dokumen Tugas Akhir. Aplikasi ini diuji dengan menggunakan metode *Black Box Testing*. Setelah diuji, aplikasi ini telah di rilis dan telah digunakan dalam produksi.

Kata kunci: *gateway*, *message routing*, *message broker* , *Store and forward*

ABSTRACT

Store and forward is one of the messaging algorithms, where message sent by sender of the message (sender) stored in advance before it is forwarded to the message receiver (receiver). Algorithms store and forward could handle obstacles such as unavailability of the destination or destinations are not active. Store and forward handles missing message during the delivery of message process takes place, missing message which is a message sent by the sender but not received by receiver. Messages will be sent through gateway and the gateway will ensure the availability of destination and forward the message to the destination. The purpose of this final project is to analyze, design and implement message routing algorithm using store and forward. The approach taken during the work is studying the message routing, the concept of store and forward algorithm, having discussion about message routing from various forums of the message broker and implement store-and-forward to handle problem where message is lost when being sent, and development using Spring Java Framework. The results of this final project are analysis and design of store-and-forward message routing, implementation and development application using Spring Java Framework and document of Final Project. The application has been tested using black box testing and has been released to production.

Keywords: gateway, message routing, message brokers, Store and Forward



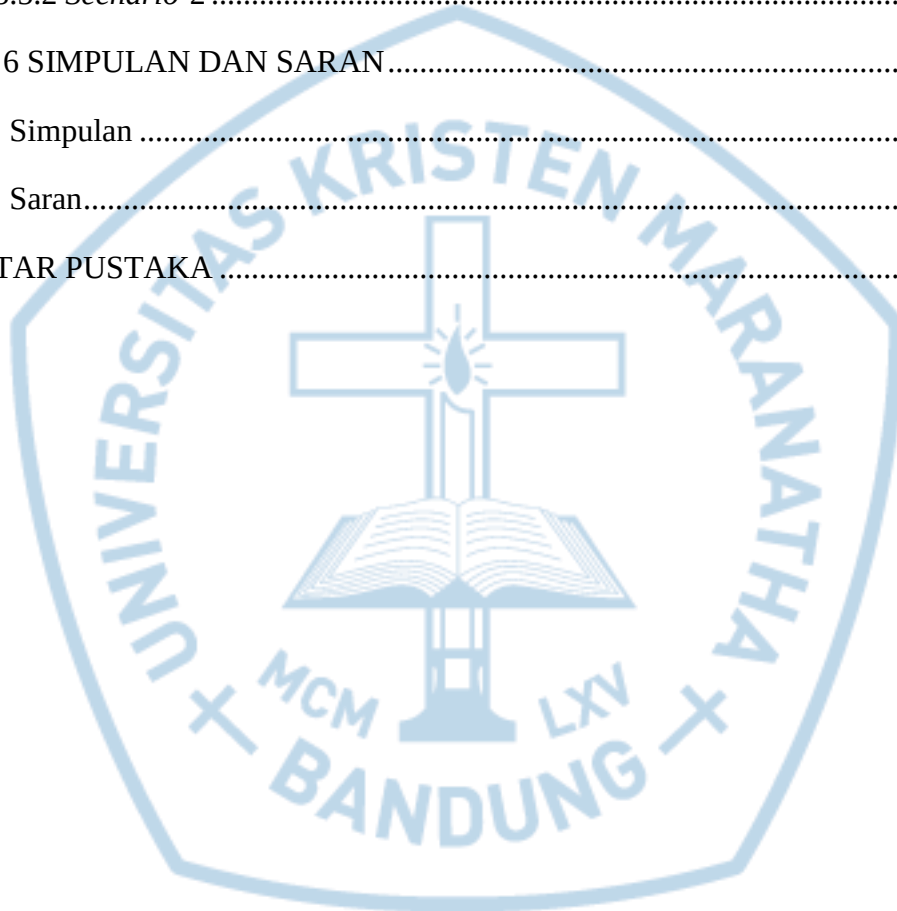
DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS LAPORAN PENELITIAN.....	ii
PERNYATAAN PUBLIKASI LAPORAN PENELITIAN	iii
PRAKATA.....	iv
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
DAFTAR NOTASI/ LAMBANG.....	xiv
DAFTAR SINGKATAN	xvii
DAFTAR ISTILAH	xviii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Pembahasan	3
1.4 Ruang Lingkup.....	4
1.5 Sumber Data.....	5
1.6 Sistematika Penyajian	5
BAB 2 KAJIAN TEORI	7
2.1 <i>Message Broker</i>	7
2.2 <i>Store and forward Concept</i>	8
2.3 <i>High Availability Concept</i>	8
2.4 <i>Design Store and forward Message Routing</i>	10

2.5 Synchronous dan Asynchronous Message Consumer	10
2.5.1 Synchronous Message Consumer.....	10
2.5.2 Asynchronous Message Consumer.....	11
2.6 Spring Framework	11
2.7 Spring JMS Message Converter.....	12
2.8 Apache ActiveMQ.....	12
2.9 Queue	14
2.10 Konsep Message Queuing	14
2.11 Modeler – Viewer – Controller Concept.....	15
2.12 Java	16
2.13 UML (The Unified Modeling Language)	16
2.14 RMI (Remote Method Invocation).....	17
BAB 3 ANALISIS DAN RANCANGAN SISTEM.....	18
3.1 Analisis Sistem.....	18
3.2 Specification.....	18
3.2.1 Functional Specification	18
3.2.2 Non-Functional Specification	19
3.3 Design	20
3.3.1 Function Send Message.....	21
3.3.2 Function Receive Message.....	21
3.3.3 Function Store Message	22
3.3.4 Function Scheduler	22
3.3.5 Function Check Availability.....	23
3.3.6 Function Pull and Forward Message	23
3.3.7 Function Print pada RMI Server	24
3.4 UML Diagram.....	25

3.4.1 Use Case Diagram	25
3.4.2 Class Diagram	26
3.4.3 Activity Diagram	28
BAB 4 IMPLEMENTASI.....	29
4.1 Library and Dependencies	29
4.1.1 Eclipse and Java Dependencies	29
4.1.2 Maven Dependencies	30
4.2 Implementasi Fungsi	31
4.2.1 Function Send Message.....	31
4.2.2 Function Receive Message	33
4.2.3 Function Store Message	34
4.2.4 Function Scheduler	35
4.2.5 Function Check Destination Availability.....	36
4.2.6 Function Pull and Forward Message	37
4.2.7 Function Print Message	39
BAB 5 UJI COBA HASIL PENELITIAN.....	41
5.1 Pengujian Blax-Box.....	41
5.2 Test Preparation	41
5.2.1 Prosedural Preparation	41
5.2.2 Hardware dan Network Preparation	41
5.2.3 Software Preparation	41
5.3 Test Plan and Identification	42
5.4 Test Script and Result	43
5.4.1 Test Script Butir-Uji-1	43
5.4.2 Test Script Butir-Uji-2	44
5.4.3 Test Script Butir-Uji-3	45

5.4.4 <i>Test Script</i> Butir-Uji-4	46
5.4.5 <i>Test Script</i> Butir-Uji-5	47
5.4.6 <i>Test Script</i> Butir-Uji-6	48
5.4.7 <i>Test Script</i> Butir-Uji-7	50
5.5 <i>Test Summary Result and History</i>	51
5.5.1 <i>Scenario-1</i>	51
5.5.2 <i>Scenario-2</i>	53
BAB 6 SIMPULAN DAN SARAN	55
6.1 Simpulán	55
6.2 Sarán.....	55
DAFTAR PUSTAKA	1



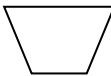
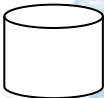
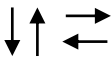
DAFTAR GAMBAR

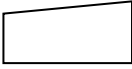
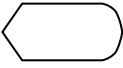
Gambar 2. 1 Ilustrasi Pekerjaan <i>Message Broker</i>	7
Gambar 2. 2 <i>Design Store and forward Message Routing</i>	10
Gambar 2. 3 <i>MVC Architecture</i>	15
 Gambar 3. 1 <i>Design Store and Forward Message Routing</i>	20
Gambar 3. 2 <i>Use Case Diagram Sender</i>	25
Gambar 3. 3 <i>Use Case Diagram SAF-Gateway</i>	25
Gambar 3. 4 <i>Class Diagram SAF-Gateway</i>	26
Gambar 3. 5 <i>Class Diagram untuk Simulator Sender</i>	27
Gambar 3. 6 <i>Activity Diagram Store and Forward Message Routing</i>	28
 Gambar 4. 1 <i>Library</i>	29
Gambar 4. 2 <i>Library Maven Dependencies</i>	31
Gambar 4. 3 Implementasi Fungsi <i>Send Message</i>	33
Gambar 4. 4 Implementasi Fungsi <i>Receive Message</i>	34
Gambar 4. 5 Implementasi Fungsi <i>Store Message</i>	35
Gambar 4. 6 Implementasi Fungsi <i>Scheduler</i>	36
Gambar 4. 7 Implementasi Fungsi <i>Check Destination Availability</i>	36
Gambar 4. 8 Implementasi Fungsi <i>Pull and Forward Message</i>	39
Gambar 4. 9 Implementasi Fungsi <i>Print Message</i>	40
 Gambar 5. 1 <i>Pending Message pada queueSender</i>	44
Gambar 5. 2 <i>Message Dequeued pada queueSender</i>	45
Gambar 5. 3 Notifikasi Destinasi Tidak Terkoneksi	48
Gambar 5. 4 <i>Message Dequeued pada queueStore sama dengan queueSender</i>	50
Gambar 5. 5 Pesan Ditampilkan di <i>Console</i>	50
Gambar 5. 6 <i>Message Dequeued</i>	50
Gambar 5. 7 <i>Server RMI Siap terkoneksi</i>	51

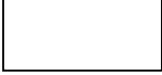
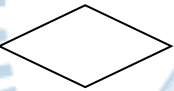
DAFTAR TABEL

Tabel 1. 1 Batasan Fungsionalitas.....	4
Tabel 2. 1Tanggung jawab dan kerja sama <i>Message Broker</i>	8
Tabel 3. 1 <i>Functional Specification</i>	18
Tabel 3. 2 <i>Function Send Message</i>	21
Tabel 3. 3 <i>Function Receive Message</i>	21
Tabel 3. 4 <i>Function Store Message</i>	22
Tabel 3. 5 <i>Function Scheduler</i>	22
Tabel 3. 6 <i>Function Check Availability</i>	23
Tabel 3. 7 <i>Function Pull and Forward Message</i>	23
Tabel 3. 8 <i>Function Print pada RMI Server</i>	24
Tabel 5. 1 <i>Test Plan</i>	42
Tabel 5. 2 Test Script Butir Uji 1	43
Tabel 5. 3 Test Script Butir Uji-2.....	44
Tabel 5. 4 Test Script Butir Uji-3.....	45
Tabel 5. 5 Test Script Butir Uji-4.....	46
Tabel 5. 6 Test script Butir Uji-5	47
Tabel 5. 7 Test script Butir Uji-6	48
Tabel 5. 8 Test script Butir Uji-7	50
Tabel 5. 9 Rekapulasi <i>Testing</i> Scenario-1.....	51
Tabel 5. 10 Rekapulasi <i>Testing</i> Scenario-2.....	53

DAFTAR NOTASI/ LAMBANG

Jenis	Notasi/Lambang	Nama	Arti
<i>Flowmap</i>		Dokumen	Digunakan untuk menggambarkan semua jenis dokumen yang merupakan formulir yang digunakan untuk merekam data keluarga.
		Proses <i>online</i> komputer	Kegiatan proses dari operasi program komputer.
		Proses manual	Proses manual pada <i>flowmap</i> .
		<i>File Harddisk</i>	Media penyimpanan dari proses <i>entry</i> data dan proses komputerisasi.
		Arsip Permanen	Tempat penyimpanan dokumen yang tidak akan diproses lagi.
		Arsip Sementara	Tempat penyimpanan data berupa arsip.
		Garis Alir	Arus data.

		<i>Keyboard</i>	Proses penyimpanan menggunakan keyboard.
		<i>Terminator</i>	Untuk mulai atau selesai
		<i>On-page connector</i>	Penghubung pada halaman yang sama.
		<i>Off-page connector</i>	Penghubung pada halaman yang berbeda.
		Pita magnetik	Untuk menggambarkan arsip komputer yang berbentuk pita magnetik.
		<i>Online Storage</i>	Untuk menggambarkan arsip komputer yang berbentuk online (di dalam memori komputer).
		<i>Decision</i>	Untuk menggambarkan keputusan yang harus dibuat dalam proses pengolahan data.
		<i>Predefined Process</i>	Lambang fungsi atau sub fungsi.
		Display	Lambang untuk mencetak keluaran dalam layar <i>monitor</i> .
		<i>Input/output</i>	Menyatakan proses <i>input/output</i> tanpa

			tergantung jenis peralatannya.
		Entitas	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data
		Garis Penghubung	Penghubung antara relasi dan entitas di mana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah Nama_atribut Nama_kunci primer pemakaian
		Relasi	Relasi yang menghubungkan antarentitas; biasanya diawali dengan kata kerja
		Attribut	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas

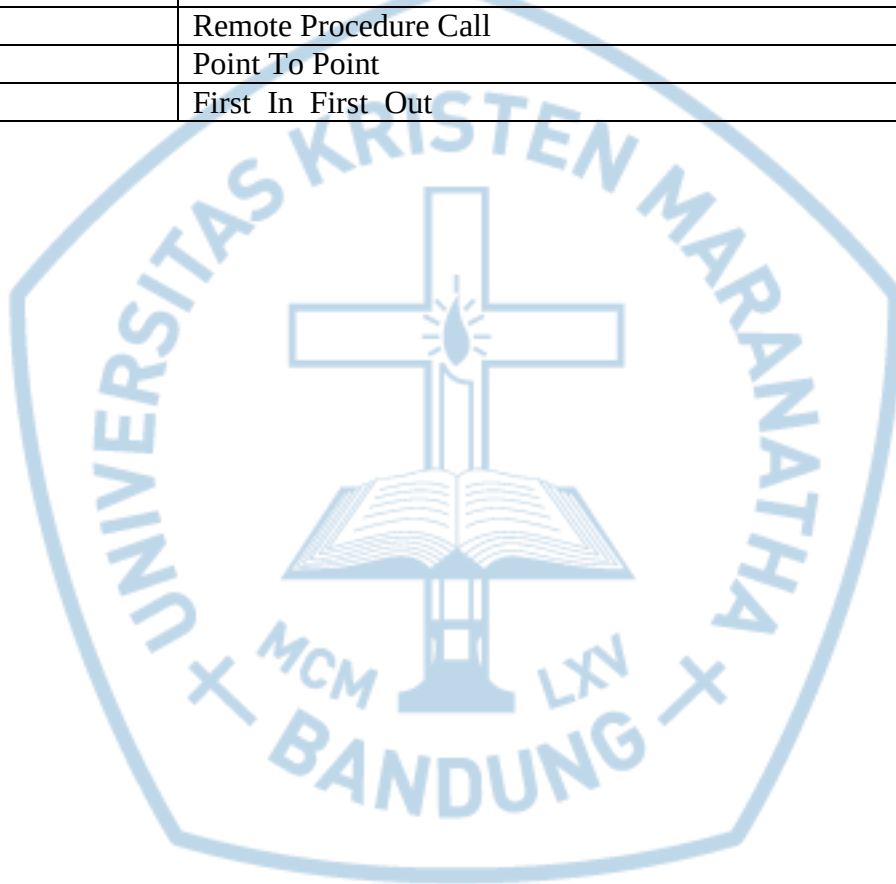
Referensi:

Notasi/ Lambang Flowmapdari Pressman[14]

Notasi/ Lambang ERD dari Janner Sirmamata [15]

DAFTAR SINGKATAN

ERD	Entity Relationship Diagram
RPL	Rekayasa Perangkat Lunak
UML	Unified Modelling Language
MB	Message Broker
HA	High Availability
SAF	Store and Forward
MVC	Modeler Viewer Controller
JMS	Java Message Service
RMI	Remote Method Invocation
RPC	Remote Procedure Call
PTP	Point To Point
FIFO	First In First Out



DAFTAR ISTILAH

Localhost	Nama standar yang diberikan sebagai alamat <i>loopback network interface</i>
Adaptive routing	Mengacu pada algoritma di mana <i>decisions</i> dibuat pada saat <i>runtime</i> di <i>router</i> menggunakan informasi lokal untuk menentukan <i>hop</i> berikutnya dari suatu paket yang telah ditentukan.
Deterministic algorithm	Pengrutean berdasarkan informasi yang tersedia sebelum algoritma dimulai; ini berarti bahwa semua rute paket ditentukan sebelum <i>runtime</i> dan masing-masing paket hanya memiliki satu rute per sumber-destinasi.
Quasi-static routing algorithm	Mirip dengan <i>Deterministic algorithm</i> , kecuali akan <i>generate fixed parameters</i> untuk pengrutean secara berkala. Hal ini memberikan algoritma dapat mengadaptasi kondisi jaringan.
Minimal routing algorithms	Algoritma yang tidak pernah membiarkan paket untuk dirutekan jauh dari tujuan.
Non-minimal routers	<i>Can allow hops in any direction</i>
Deadlock	<i>Deadlock</i> digunakan untuk merujuk kepada keadaan jaringan di mana paket tidak bisa membuat <i>progress</i> dalam jaringan karena <i>a cyclic wait dependency</i> .
Livelock	<i>Livelock</i> mengacu pada situasi di mana sebuah paket telah dimasukkan ke jaringan dan terus dirutekan, tetapi tidak pernah mencapai tujuannya
Switching	<i>Switching</i> digunakan untuk menunjukkan mekanisme yang digunakan <i>router</i> untuk memindahkan paket dari <i>input</i> ke <i>output</i> .
Routing	Mengindikasikan proses pemilihan <i>path</i> yang akan diikuti oleh paket melalui jaringan.
Message Routing	Proses penyampaian pesan melalui <i>gateway</i> dari

	<i>sender ke receiver</i>
RMI (Remote Method Invocation)	RMI adalah sebuah teknik pemanggilan method remote yang secara umum lebih baik dari RPC. RMI menggunakan paradigma pemrograman berorientasi obyek. RMI memungkinkan kita untuk mengirim obyek sebagai parameter dari <i>remote method</i> . Dengan diperbolehkannya program Java memanggil <i>method</i> pada <i>remote</i> obyek, RMI membuat pengguna dapat mengembangkan aplikasi Java yang terdistribusi pada jaringan.

