

GARUDAHOKI WhatsApp Bot - Testing Guide

Panduan Lengkap Testing Bot

Overview

Panduan ini akan membantu Anda melakukan testing menyeluruh terhadap bot WhatsApp GARUDAHOKI untuk memastikan semua fitur berfungsi dengan baik sebelum go-live.

Langkah 1: Pre-Testing Checklist

A. Verifikasi Environment Setup

1. Check Application Status

```
```bash
Verify bot application running
pm2 status garudahoki-bot

Check application health
curl https://yourdomain.com/api/health
Expected: {"status": "OK", "timestamp": "..."}
```
```

1. Verify Database Connection

```
```bash
Test database connectivity
psql -h localhost -U garudahoki_user -d garudahoki_bot -c "SELECT 1;"

Check tables exist
psql -h localhost -U garudahoki_user -d garudahoki_bot -c "\dt"
```
```

1. Check Environment Variables

```
bash
# Verify critical environment variables
env | grep -E "(WHATSAPP|LIVECHAT|ADMIN)" | grep -v PASSWORD
```

B. Verify External Integrations

1. WhatsApp API Connection

```
bash
# Test WhatsApp API (example for SleekFlow)
curl -X GET "https://api.sleekflow.io/v1/me" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

2. LiveChatInc API Connection

```
bash
# Test LiveChat API
curl -X GET "https://api.livechatinc.com/v3.5/configuration/license" \
-H "Authorization: Bearer YOUR_PAT"
```

Langkah 2: Testing Bot Core Functionality

A. Test Welcome Message & Branding

1. Test Initial Contact

...

Action: Kirim pesan "Hi" ke nomor WhatsApp bot

Expected Response:

GARUDAHOKI - Selamat Datang di Livechat GARUDAHOKI Situs Judi Slot Online Paling Gacor Di Indonesia.

LINK ALTERNATIF GARUDAHOKI:

- <https://garuda-beriman.com>
- <https://linklist.bio/login-garudahoki>
- <https://tnpcnewsletter.com/>
- <https://sildenafilliq.com/>
- <https://amandaonline.com>

Download Segera APK GARUDAHOKI!

[Time-based greeting], apakah ada yang bisa kami bantu bosku sayang?

...

1. Test Time-Based Greetings

...

Test Cases:

- 08:00 WIB: Should show "Selamat Pagi"
- 12:00 WIB: Should show "Selamat Siang"
- 16:00 WIB: Should show "Selamat Sore"
- 20:00 WIB: Should show "Selamat Malam"

Note: Adjust server time to test different periods

...

B. Test Menu System

1. Test Main Menu

...

Action: Setelah welcome message, bot harus menampilkan menu

Expected Response:

Silakan pilih menu:

- a) Informasi Umum
- b) Bicara dengan Agen

...

1. Test Informasi Umum Menu

...

Action: Reply dengan "a" atau "Informasi"

Expected Response:

Pilih informasi yang Anda butuhkan:

- 1) Bonus Hari Ini

2) Pola Harian

...

1. Test Bonus Hari Ini

...

Action: Reply dengan "1" atau "Bonus"

Expected Response: [Dynamic content dari admin panel]

Default: "Bonus hari ini sedang diupdate, silakan cek kembali nanti."

...

1. Test Pola Harian

...

Action: Reply dengan "2" atau "Pola"

Expected Response: [Dynamic content dari admin panel]

Default: "Pola harian sedang diupdate, silakan cek kembali nanti."

...

C. Test Agent Handover

1. Test Agent Request

...

Action: Reply dengan "b" atau "Agen"

Expected Response:

"Baik, silakan deskripsikan kendala yang Anda alami secara singkat."

...

1. Test Handover Process

...

Action: Reply dengan deskripsi masalah, contoh: "Saya tidak bisa login ke akun"

Expected Response:

"Terima kasih, pesan Anda telah kami teruskan. Mohon tunggu, Anda akan segera terhubung langsung dengan agen kami."

Verification:

- Check LiveChatInc dashboard untuk chat baru
- Verify chat assigned ke agen Beatrice
- Check admin panel untuk handover log

...

Langkah 3: Testing Content Filter

A. Test Profanity Filter

1. Test First Violation

...

Action: Kirim pesan dengan kata kasar: "anjing"

Expected Response:

"Bosku, mohon gunakan bahasa yang baik yah bos ku, jika ingin di layani dengan baik. Terimakasih

^^"

Verification:

- Check database: user_violations table should have 1 entry
 - User should NOT be blocked yet
- ...

1. Test Second Violation

...

Action: Kirim pesan dengan kata kasar lagi: "tai"

Expected Response:

"Bosku, mohon gunakan bahasa yang baik yah bos ku, jika ingin di layani dengan baik. Terimakasih
^^"

Verification:

- Check database: user_violations table should have 2 entries
 - User should NOT be blocked yet
- ...

1. Test Third Violation (Auto-Block)

...

Action: Kirim pesan dengan kata kasar ketiga: "babi"

Expected Response:

"Bosku, mohon gunakan bahasa yang baik yah bos ku, jika ingin di layani dengan baik. Terimakasih
^^"

Then bot should:

- End chat session
- Block user permanently
- Stop responding to messages from this number

Verification:

- Check database: blocked_numbers table should have entry
 - Check admin panel: user should appear in blocked list
 - Test: Send another message, should get no response
- ...

B. Test Blocked User Behavior

1. Test Blocked User Messages

...

Action: Dari nomor yang sudah diblokir, kirim pesan apapun

Expected Response: No response (bot ignores completely)

Verification:

- Check logs: should show "Message from blocked number ignored"
 - No database entries should be created for new messages
- ...

Langkah 4: Testing Complaint Auto-Response

A. Test Complaint Keywords

1. Test “kalah” keyword

...

Action: Kirim pesan: “Saya kalah terus nih”

Expected Response:

“Setiap Permainan Ada Kalah dan Pasti Ada Menangnya Bosku 😊

Semakin Banyak Mencoba Semakin Banyak Kesempatan Untuk Menang.

Kami disini hanya bisa memproses transaksi bosku saja^^ untuk menang atau kalah kembali di hoki bosku yah 😊

Bermain Dengan Sabar, Fokus dan Optimis Terhadap Kemenangan.

Salam JP, GARUDAHOKI 🙏😊”

...

1. Test Other Complaint Keywords

...

Test Cases:

- “sedot wc” → Should trigger complaint response
- “rungkat” → Should trigger complaint response
- “bangsat” → Should trigger complaint response
- “parah” → Should trigger complaint response
- “mengecewakan” → Should trigger complaint response

All should return the same complaint response above

...

B. Test Mixed Messages

1. Test Complaint + Normal Flow

...

Action: Kirim “Ini parah banget”

Expected: Complaint auto-response (no menu)

Action: Kirim “Hi” lagi

Expected: Normal welcome message dengan menu

...

Langkah 5: Testing Admin Panel

A. Test Admin Login

1. Test Login Page

...

Action: Buka <https://yourdomain.com/admin>

Expected: Login form dengan GARUDAHOKI branding

...

1. Test Authentication

Test Cases:

- Wrong username: Should show error
- Wrong password: Should show error
- Correct credentials: Should redirect to dashboard

B. Test Dashboard Features

1. Test Dynamic Content Management

...

Action: Login → Navigate to “Content Management”

Test Cases:

- Update “Bonus Hari Ini” content
- Update “Pola Harian” content
- Save changes
- Verify changes reflected in bot responses

...

1. Test Blocked Numbers Management

...

Action: Navigate to “Blocked Numbers”

Expected Features:

- List of blocked numbers
- Unblock functionality
- Block reason/date
- Search/filter capability

...

1. Test Statistics Dashboard

...

Action: Navigate to “Statistics”

Expected Metrics:

- Total chats today/week/month
- Total violations
- Total blocked numbers
- Handovers to LiveChat
- Response time metrics

...

1. Test Chat Logs

...

Action: Navigate to “Chat Logs”

Expected Features:

- Recent chat history
- Search by phone number
- Filter by date range
- Export functionality

...

Langkah 6: Integration Testing

A. Test WhatsApp Provider Integration

1. Test Message Delivery

...

Action: Send various message types

Test Cases:

- Text messages
 - Messages with emojis
 - Long messages (>1000 characters)
 - Messages with special characters
- ...

1. Test Webhook Reliability

...

Action: Send rapid messages (stress test)

Expected: All messages processed in order

Verification: Check logs for any dropped messages

...

B. Test LiveChatInc Integration

1. Test Chat Creation

...

Action: Trigger agent handover

Verification:

- New chat appears in LiveChatInc dashboard
 - Chat contains customer context
 - Chat assigned to correct agent (Beatrice)
- ...

1. Test Bidirectional Communication

...

Action: Agent replies in LiveChatInc

Expected: Reply should reach customer via WhatsApp

(Note: This requires full integration setup)

...

Langkah 7: Performance Testing

A. Load Testing

1. Test Concurrent Users

```
bash
# Simulate multiple users
for i in {1..10}; do
  curl -X POST "https://yourdomain.com/webhook/whatsapp" \
    -H "Content-Type: application/json" \
```

```
-d "{\"from\":\"62812345678$i\",\"body\":\"Hi\"}" &
done
```

2. Test Database Performance

```
```sql
- Check query performance
EXPLAIN ANALYZE SELECT * FROM chat_sessions WHERE created_at > NOW() - INTERVAL '1 day';

- Check database size
SELECT pg_size_pretty(pg_database_size('garudahoki_bot'));
```
```

B. Memory & CPU Testing

1. Monitor Resource Usage

```
```bash
Monitor during testing
pm2 monit

Check memory usage
ps aux | grep node

Check disk usage
df -h
```
```



Langkah 8: Error Handling Testing

A. Test Network Failures

1. Test WhatsApp API Downtime

```
```
Action: Temporarily block WhatsApp API endpoint
Expected: Bot should handle gracefully, log errors, retry
```
```

1. Test Database Connection Loss

```
```
Action: Temporarily stop PostgreSQL
Expected: Bot should handle gracefully, show maintenance message
```
```

B. Test Invalid Inputs

1. Test Malformed Webhooks

```
```bash
Send invalid JSON
curl -X POST "https://yourdomain.com/webhook/whatsapp" \
-H "Content-Type: application/json" \
-d "invalid json"
```

# Expected: 400 error, no crash

...

### 1. Test SQL Injection Attempts

...

Action: Send message with SQL injection patterns

Example: “'; DROP TABLE users; –”

Expected: Message processed normally, no database impact

...



## Langkah 9: User Acceptance Testing

### A. End-to-End User Journey

#### 1. New Customer Journey

...

Scenario: First-time customer interaction

Steps:

1. Customer sends “Hi” → Gets welcome message
2. Customer selects “a” → Gets info menu
3. Customer selects “1” → Gets bonus info
4. Customer selects “b” → Requests agent
5. Customer describes issue → Gets handover confirmation

Success Criteria: Smooth flow, appropriate responses

...

#### 1. Returning Customer Journey

...

Scenario: Customer returns after previous interaction

Steps:

1. Customer sends message → Gets welcome (not first-time)
2. Bot recognizes returning customer
3. Provides personalized experience

Success Criteria: Context awareness, personalization

...

### B. Edge Case Testing

#### 1. Test Unusual Inputs

Test Cases:

- Empty messages
- Only emojis
- Very long messages
- Messages in different languages
- Numbers only
- Special characters

#### 2. Test Timing Edge Cases

Test Cases:

- Messages sent exactly at midnight (greeting change)
- Rapid consecutive messages
- Messages during server restart
- Messages during maintenance

---

## Langkah 10: Testing Checklist

---

### A. Pre-Launch Checklist

- ☐ Welcome message displays correctly with all links
- ☐ Time-based greetings work for all time periods
- ☐ Menu system responds to all input variations
- ☐ Dynamic content updates from admin panel
- ☐ Agent handover creates LiveChat session
- ☐ Profanity filter blocks after 3 violations
- ☐ Complaint auto-responses work for all keywords
- ☐ Admin panel login and all features functional
- ☐ Database logging all interactions correctly
- ☐ Error handling graceful for all failure scenarios
- ☐ Performance acceptable under load
- ☐ Security measures in place
- ☐ Backup and recovery procedures tested

### B. Go-Live Checklist

- ☐ Production environment configured
- ☐ SSL certificates valid
- ☐ Domain pointing correctly
- ☐ WhatsApp Business API verified and live
- ☐ LiveChatInc integration active
- ☐ Monitoring and alerting configured
- ☐ Admin credentials secure
- ☐ Documentation complete
- ☐ Team trained on admin panel
- ☐ Support procedures established

---

## Troubleshooting Common Test Issues

---

### Issue 1: Bot Not Responding

#### Symptoms:

- Messages sent but no response
- Webhook receiving messages but bot silent

#### Debug Steps:

```
Check application status
pm2 status garudahoki-bot

Check logs for errors
pm2 logs garudahoki-bot --lines 50

Test webhook manually
curl -X POST "https://yourdomain.com/webhook/whatsapp" \
 -H "Content-Type: application/json" \
 -d '{"from":"628123456789","body":"test"}'
```

## Issue 2: Database Errors

### Symptoms:

- Admin panel not loading
- Chat history not saving

### Debug Steps:

```
Test database connection
psql -h localhost -U garudahoki_user -d garudahoki_bot -c "SELECT 1;"

Check database logs
tail -f /var/log/postgresql/postgresql-*.log

Verify table structure
psql -h localhost -U garudahoki_user -d garudahoki_bot -c "\d+ chat_sessions"
```

## Issue 3: Integration Failures

### Symptoms:

- LiveChat handover not working
- WhatsApp messages not sending

### Debug Steps:

```
Test API connections
curl -X GET "https://api.livechatinc.com/v3.5/configuration/license" \
 -H "Authorization: Bearer YOUR_PAT"

Check environment variables
env | grep -E "(WHATSAPP|LIVECHAT)"

Verify webhook configurations
```

---

## Support & Next Steps

### After Testing Complete

#### 1. Document Test Results

- Create test report with all results
- Note any issues found and resolutions
- Document performance benchmarks

## 2. Plan Go-Live

- Schedule deployment window
- Prepare rollback plan
- Set up monitoring alerts
- Train support team

## 3. Post-Launch Monitoring

- Monitor bot performance first 24 hours
- Check error rates and response times
- Gather user feedback
- Plan iterative improvements

## Emergency Contacts

- **Technical Issues:** Check application logs first
- **WhatsApp API Issues:** Contact provider support
- **LiveChatInc Issues:** Contact LiveChat support
- **Hosting Issues:** Contact Namecheap support

---

🎉 **Testing Complete! Bot WhatsApp GARUDAHOKI siap untuk go-live setelah semua test cases passed.**

---