

Contents

MAINTENANCE GUIDE	2
Document Information	2
Overview	2
MAINTENANCE SCHEDULE	2
Daily Tasks (Setiap Hari)	2
Weekly Tasks (Setiap Minggu)	2
Monthly Tasks (Setiap Bulan)	2
1. SYSTEM MONITORING	3
1.1 Server Health Check	3
1.2 Application Health	3
1.3 Log Monitoring	4
2. DATABASE MAINTENANCE	4
2.1 Daily Backup Verification	4
2.2 Database Performance Monitoring	5
2.3 Database Cleanup (Monthly)	6
3. CACHE MANAGEMENT	6
3.1 Clear Application Cache	6
3.2 Redis Cache (If Enabled)	7
4. FILE SYSTEM MAINTENANCE	7
4.1 Disk Space Management	7
4.2 File Permission Check	8
5. PERFORMANCE OPTIMIZATION	8
5.1 PHP-FPM Tuning	8
5.2 Nginx Optimization	9
5.3 Database Indexing	9
6. SECURITY MAINTENANCE	9
6.1 SSL Certificate Renewal	9
6.2 Security Updates	10
6.3 Audit Log Review	10
7. QUEUE & JOBS MANAGEMENT	11
7.1 Queue Worker Status	11
7.2 Failed Jobs Management	11
8. MONITORING & ALERTS	12
8.1 Application Metrics	12
8.2 Uptime Monitoring	12
8.3 Alert Configuration	12
9. TROUBLESHOOTING GUIDE	12
9.1 Application Not Accessible	12
9.2 Slow Performance	13
9.3 Database Connection Error	14
10. CHECKLISTS	14
10.1 Daily Checklist	14
10.2 Weekly Checklist	15
10.3 Monthly Checklist	15
11. ESCALATION CONTACTS	15
12. REFERENCES	15

MAINTENANCE GUIDE

Document Information

Application: Koperasi Karyawan SKF

Version: 2.0.0

Maintainer: IT Team

Last Updated: 17 January 2026

Overview

Dokumen ini berisi panduan lengkap untuk **maintenance & monitoring** aplikasi Koperasi Karyawan SKF dalam operasional harian, mingguan, dan bulanan.

Target Audience: - System Administrator - DevOps Engineer - IT Support Team

MAINTENANCE SCHEDULE

Daily Tasks (Setiap Hari)

Time	Task	Duration	Priority
08:00	Check system uptime & accessibility	5 min	P0
08:10	Review overnight backup status	10 min	P0
08:30	Check error logs for critical issues	15 min	P0
09:00	Monitor disk space usage	5 min	P1
17:00	Review day's audit logs	10 min	P1
17:30	Check pending notifications queue	5 min	P2

Total Daily: ~50 minutes

Weekly Tasks (Setiap Minggu)

Day	Task	Duration
Monday	Review security logs & access patterns	30 min
Tuesday	Database performance analysis	45 min
Wednesday	Update system dependencies (check)	20 min
Thursday	Backup verification (test restore)	60 min
Friday	Weekly status report	30 min
Saturday	System health check comprehensive	45 min
Sunday	Full system backup (automated + verify)	15 min

Total Weekly: ~4 hours

Monthly Tasks (Setiap Bulan)

Task	Schedule	Duration
Security patch review & apply	Week 1	2 hours
Database optimization & cleanup	Week 2	1.5 hours
SSL certificate renewal check	Week 2	15 min
Audit log archive	Last day of month	30 min
Performance report generation	Last day of month	1 hour
Disaster recovery drill	Last Sunday	2 hours

Total Monthly: ~7 hours

1. SYSTEM MONITORING

1.1 Server Health Check

Command:

```
# Check system resources
cd /var/www/koperasi
./scripts/health-check.sh
```

Manual Checks:

```
# CPU & Memory
htop
free -h

# Disk space (Alert if > 80%)
df -h
```

```
# Running services
systemctl status nginx
systemctl status php8.2-fpm
systemctl status mysql
systemctl status redis-server
systemctl status supervisor
```

```
# Network connectivity
ping -c 3 google.com
```

Expected Output:

```
CPU usage: < 70%
Memory usage: < 80%
Disk usage: < 80%
All services: active (running)
Network: accessible
```

If Issues: - CPU > 80%: Check PHP-FPM pool, identify slow queries - Memory > 90%: Restart PHP-FPM, clear cache - Disk > 85%: Clean old logs, backups

1.2 Application Health

URL Health Check:

```

# Via curl
curl -I https://kopkarskf.com

# Expected: HTTP/2 200

Laravel Health:

cd /var/www/koperasi

# Check application status
php artisan inspire # Should return inspirational quote

# Check database connection
php artisan tinker
>>> DB::connection()->getPdo(); exit;

# Check queue workers
php artisan queue:work --once

```

1.3 Log Monitoring

Location:

```
/var/www/koperasi/storage/logs/
    laravel.log      # Application logs
    audit.log        # Audit trail
    nginx-access.log # Web server access
    nginx-error.log  # Web server errors
    php-fpm.log      # PHP errors
```

Daily Check:

```

# Check for errors in last 24 hours
cd /var/www/koperasi/storage/logs

# Critical errors
grep -i "error\|exception\|fatal" laravel.log | tail -50

# Failed logins
grep "Failed login" audit.log | tail -20

# 500 errors
grep " 500 " nginx-access.log | tail -20

```

Alert Triggers: - > 10 errors/hour in laravel.log - > 5 failed logins from same IP - > 5 HTTP 500 errors/hour

Action: - Investigate error context - Check stack trace - Apply fix - Monitor for recurrence

2. DATABASE MAINTENANCE

2.1 Daily Backup Verification

Location:

```
/var/www/koperasi/storage/app/backups/  
Format: koperasi_YYYYMMDD_HHMMSS.sql.gz
```

Check:

```
# Verify backup exists (created at 02:00 daily)  
ls -lh /var/www/koperasi/storage/app/backups/ | grep $(date +%Y%m%d)  
  
# Check file size (should be > 10MB)  
du -h /var/www/koperasi/storage/app/backups/koperasi_${(date +%Y%m%d)}*.sql.gz  
  
# Test backup integrity  
gunzip -t /var/www/koperasi/storage/app/backups/koperasi_${(date +%Y%m%d)}*.sql.gz  
echo $? # Should output: 0 (success)
```

If Backup Missing:

```
# Manual backup  
cd /var/www/koperasi  
php artisan backup:run
```

2.2 Database Performance Monitoring

Weekly Check:

```
mysql -u root -p koperasi  
  
-- Check database size  
SELECT  
    table_schema AS 'Database',  
    ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Size (MB)'  
FROM information_schema.tables  
WHERE table_schema = 'koperasi'  
GROUP BY table_schema;  
  
-- Check largest tables  
SELECT  
    table_name AS 'Table',  
    ROUND((data_length + index_length) / 1024 / 1024, 2) AS 'Size (MB)'  
FROM information_schema.tables  
WHERE table_schema = 'koperasi'  
ORDER BY (data_length + index_length) DESC  
LIMIT 10;  
  
-- Check slow queries (> 2 seconds)  
SELECT * FROM mysql.slow_log  
WHERE sql_text NOT LIKE '%information_schema%'  
ORDER BY query_time DESC  
LIMIT 10;  
  
-- Check table fragmentation  
SELECT  
    table_name,  
    ROUND(data_length / 1024 / 1024, 2) AS data_mb,  
    ROUND(data_free / 1024 / 1024, 2) AS free_mb,
```

```

    ROUND((data_free / data_length) * 100, 2) AS fragmentation_pct
FROM information_schema.tables
WHERE table_schema = 'koperasi'
    AND data_free > 0
ORDER BY fragmentation_pct DESC;

exit;

If Fragmentation > 10%:

mysql -u root -p koperasi

-- Optimize table
OPTIMIZE TABLE table_name;

-- Or optimize all tables
mysqlcheck -u root -p --optimize koperasi

```

2.3 Database Cleanup (Monthly)

Old Data Cleanup:

```

mysql -u root -p koperasi

-- Delete old sessions (> 7 days)
DELETE FROM sessions
WHERE last_activity < UNIX_TIMESTAMP(DATE_SUB(NOW(), INTERVAL 7 DAY));

-- Delete old failed_jobs (> 30 days)
DELETE FROM failed_jobs
WHERE failed_at < DATE_SUB(NOW(), INTERVAL 30 DAY);

-- Delete old password_reset_tokens (> 1 day)
DELETE FROM password_reset_tokens
WHERE created_at < DATE_SUB(NOW(), INTERVAL 1 DAY);

-- Archive old audit_logs (> 1 year to separate table)
INSERT INTO audit_logs_archive
SELECT * FROM audit_logs
WHERE created_at < DATE_SUB(NOW(), INTERVAL 1 YEAR);

DELETE FROM audit_logs
WHERE created_at < DATE_SUB(NOW(), INTERVAL 1 YEAR);

exit;

```

3. CACHE MANAGEMENT

3.1 Clear Application Cache

When to Clear: - After code deployment - After configuration changes - Performance degradation - Weekly maintenance

Commands:

```
cd /var/www/koperasi

# Clear all caches
php artisan cache:clear
php artisan config:clear
php artisan route:clear
php artisan view:clear

# Rebuild caches (production only)
php artisan config:cache
php artisan route:cache
php artisan view:cache

# Restart services
sudo systemctl restart php8.2-fpm
sudo systemctl reload nginx
```

3.2 Redis Cache (If Enabled)

Monitor:

```
redis-cli
```

```
# Check memory usage
INFO memory

# Check key count
DBSIZE

# Check slow commands
SLOWLOG GET 10
```

```
exit
```

Clear Redis:

```
redis-cli FLUSHALL
```

4. FILE SYSTEM MAINTENANCE

4.1 Disk Space Management

Monitor:

```
# Overall disk usage
df -h

# Find large directories
du -sh /var/www/koperasi/* | sort -hr | head -10

# Find large files
find /var/www/koperasi -type f -size +100M -exec ls -lh {} \;

Cleanup Actions:
```

1. Old Log Files (> 30 days):

```
find /var/www/koperasi/storage/logs -name "*.log" -mtime +30 -delete
```

2. Old Backup Files (> 90 days):

```
find /var/www/koperasi/storage/app/backups -name "*.sql.gz" -mtime +90 -delete
```

3. Temporary Files:

Session files

```
find /var/www/koperasi/storage/framework/sessions -type f -mtime +7 -delete
```

Cache files

```
find /var/www/koperasi/storage/framework/cache -type f -mtime +7 -delete
```

Temporary uploads

```
find /var/www/koperasi/storage/app/tmp -type f -mtime +1 -delete
```

4. Compress Old Logs:

```
cd /var/www/koperasi/storage/logs  
gzip *.log.$(date -d "7 days ago" +%Y-%m-%d)
```

4.2 File Permission Check

Verify Permissions:

```
cd /var/www/koperasi
```

Check ownership

```
ls -la storage bootstrap/cache
```

Expected:

```
# drwxrwxr-x www-data www-data storage  
# drwxrwxr-x www-data www-data bootstrap/cache
```

Fix if needed

```
sudo chown -R www-data:www-data storage bootstrap/cache  
sudo chmod -R 775 storage bootstrap/cache
```

5. PERFORMANCE OPTIMIZATION

5.1 PHP-FPM Tuning

Check Current Settings:

```
sudo cat /etc/php/8.2/fpm/pool.d/www.conf | grep -E "pm.max_children|pm.start_servers|pm.min_spare_servers|pm.max_requests"
```

Recommended Settings (4GB RAM server):

```
pm = dynamic  
pm.max_children = 50  
pm.start_servers = 10  
pm.min_spare_servers = 5  
pm.max_spare_servers = 20  
pm.max_requests = 500
```

Apply Changes:

```
sudo systemctl restart php8.2-fpm
```

5.2 Nginx Optimization

Enable Gzip:

```
# /etc/nginx/nginx.conf
gzip on;
gzip_types text/plain text/css application/json application/javascript text/xml application/xml;
gzip_min_length 1000;
```

Browser Cache:

```
# Static assets
location ~* \.(jpg|jpeg|png|gif|ico|css|js|svg|woff|woff2)$ {
    expires 365d;
    add_header Cache-Control "public, immutable";
}
```

Reload Nginx:

```
sudo nginx -t
sudo systemctl reload nginx
```

5.3 Database Indexing

Check Missing Indexes:

```
-- Find queries without indexes
SELECT * FROM mysql.slow_log
WHERE sql_text NOT LIKE '%FROM information_schema%'
ORDER BY query_time DESC
LIMIT 20;
```

Add Indexes (Example):

```
-- If you see frequently queried columns without index
CREATE INDEX idx_members_status ON members(status);
CREATE INDEX idx_loans_status ON loans(status, approved_at);
CREATE INDEX idx_transactions_date ON transactions(created_at);
```

6. SECURITY MAINTENANCE

6.1 SSL Certificate Renewal

Check Expiry:

```
# Check certificate expiration
sudo certbot certificates
```

```
# Expected output shows expiry date
```

Auto-Renewal (Let's Encrypt):

```
# Test renewal
sudo certbot renew --dry-run

# Cron job should be in place
cat /etc/cron.d/certbot
# Expected: 0 */12 * * * root certbot renew --quiet
```

Manual Renewal (if needed):

```
sudo certbot renew
sudo systemctl reload nginx
```

6.2 Security Updates

Check for Updates:

```
# System packages
sudo apt update
sudo apt list --upgradable

# Composer dependencies
cd /var/www/koperasi
composer outdated

# NPM dependencies
npm outdated
```

Apply Updates (Carefully):

```
# Security updates only
sudo apt upgrade -y

# Composer (test in staging first!)
composer update --with-dependencies
```

```
# NPM (test in staging first!)
npm update
npm run build
```

After Updates:

```
# Clear cache
php artisan cache:clear
php artisan config:clear

# Restart services
sudo systemctl restart php8.2-fpm
sudo systemctl reload nginx

# Verify application
curl -I https://kopkarskf.com
```

6.3 Audit Log Review

Weekly Review:

```

cd /var/www/koperasi

# Failed login attempts
php artisan tinker
>>> DB::table('audit_logs')->where('action', 'failed_login')->where('created_at', '>', now()->subDays(7))

# Unauthorized access (403)
>>> DB::table('audit_logs')->where('action', 'unauthorized')->where('created_at', '>', now()->subDays(7))

# Critical data changes
>>> DB::table('audit_logs')->whereIn('model', ['Loan', 'ShuDistribution', 'Member'])->where('created_at', '>', now()->subDays(7))

exit;

```

7. QUEUE & JOBS MANAGEMENT

7.1 Queue Worker Status

Check:

```

# Supervisor status (queue worker)
sudo supervisorctl status

# Expected: laravel-worker:* RUNNING

# Check failed jobs
mysql -u root -p koperasi -e "SELECT COUNT(*) FROM failed_jobs;"
```

Restart Workers:

```
# Graceful restart
php artisan queue:restart
```

```
# Force restart via Supervisor
sudo supervisorctl restart laravel-worker:*
```

7.2 Failed Jobs Management

View Failed Jobs:

```
cd /var/www/koperasi
php artisan queue:failed
```

Retry Failed Jobs:

```
# Retry specific job
php artisan queue:retry JOB_ID
```

```
# Retry all
php artisan queue:retry all
```

Clear Old Failed Jobs:

```
# Flush all failed jobs (CAUTION!)
php artisan queue:flush
```

8. MONITORING & ALERTS

8.1 Application Metrics

Daily Metrics to Track:

```
# Via tinker
php artisan tinker

# Today's statistics
>>> $today = today();
>>> [
    'transactions' => DB::table('transactions')->whereDate('created_at', $today)->count(),
    'new_members' => DB::table('members')->whereDate('created_at', $today)->count(),
    'loan_applications' => DB::table('loans')->whereDate('created_at', $today)->count(),
    'errors' => File::lines(storage_path('logs/laravel.log'))->filter(fn($line) => str_contains($line,
];

exit;
```

Weekly Report:

```
# Generate report
php artisan report:weekly

# Or manually create Excel report from database
```

8.2 Uptime Monitoring

External Monitoring (Recommended): - **UptimeRobot** (free): <https://uptimerobot.com> - Monitor: <https://kopkarskf.com> - Alert: Email + WhatsApp - Interval: 5 minutes - Alert if: Down for 2+ checks

Self-hosted Alternative:

```
# Simple cron job ping
crontab -e

# Add this line (every 5 minutes)
*/5 * * * * curl -f https://kopkarskf.com || echo "Site down!" | mail -s "ALERT: Site Down" admin@kopka
```

8.3 Alert Configuration

Email Alerts: - Disk space > 85% - Backup failed - Critical errors > 10/hour - Database connection failed - SSL certificate expiring < 30 days

WhatsApp Alerts (Critical): - Application down > 5 minutes - Database crash - Security breach detected

9. TROUBLESHOOTING GUIDE

9.1 Application Not Accessible

Symptoms: Site returns 502 Bad Gateway or timeout

Checklist:

```
# 1. Check Nginx
sudo systemctl status nginx
sudo nginx -t
# If error, check config:
sudo nano /etc/nginx/sites-available/koperasi

# 2. Check PHP-FPM
sudo systemctl status php8.2-fpm
# If not running:
sudo systemctl start php8.2-fpm

# 3. Check PHP-FPM socket
ls -la /var/run/php/php8.2-fpm.sock
# Should exist with proper permissions

# 4. Check error logs
sudo tail -50 /var/log/nginx/error.log
sudo tail -50 /var/log/php8.2-fpm.log

# 5. Restart all
sudo systemctl restart php8.2-fpm
sudo systemctl reload nginx
```

9.2 Slow Performance

Symptoms: Pages loading > 5 seconds

Diagnosis:

```
# 1. Check server resources
htop
# Look for high CPU/memory usage

# 2. Check database
mysql -u root -p koperasi
SELECT * FROM mysql.slow_log ORDER BY query_time DESC LIMIT 10;

# 3. Check PHP-FPM pool
sudo tail -50 /var/log/php8.2-fpm.log
# Look for: "max_children reached"

# 4. Enable Laravel debug bar (temporarily)
php artisan debugbar:enable

# 5. Check for N+1 queries
# (View page with debugbar enabled)
```

Solutions:

```
# Increase PHP-FPM children
sudo nano /etc/php/8.2/fpm/pool.d/www.conf
# pm.max_children = 50 # Increase this
```

```

# Optimize database
mysql -u root -p
OPTIMIZE TABLE table_name;

# Clear cache
php artisan cache:clear
php artisan view:clear

# Restart
sudo systemctl restart php8.2-fpm

```

9.3 Database Connection Error

Symptoms: “SQLSTATE[HY000] [2002] Connection refused”

Fix:

```

# 1. Check MySQL service
sudo systemctl status mysql
# If not running:
sudo systemctl start mysql

# 2. Check MySQL credentials
php artisan config:show database

# 3. Test connection
mysql -h 127.0.0.1 -u DB_USERNAME -p
# Enter password from .env

# 4. Check MySQL logs
sudo tail -50 /var/log/mysql/error.log

# 5. Restart MySQL if needed
sudo systemctl restart mysql

```

10. CHECKLISTS

10.1 Daily Checklist

Date: ____/____/_____
By: _____

Morning (08:00-09:00):

- Site accessible (curl test)
- Backup completed (check file)
- Error logs reviewed (< 10 errors)
- Disk space OK (< 80%)
- All services running

Evening (17:00-18:00):

- Audit logs reviewed
- Queue jobs processed
- No pending alerts

Notes: _____

10.2 Weekly Checklist

Week of: ____/____/_____
By: _____

Security logs reviewed (no anomalies)
Database performance check
Dependency updates checked
Backup restore tested
Weekly report generated
System health comprehensive

Issues Found: _____
Actions Taken: _____

10.3 Monthly Checklist

Month: _____
By: _____

Security patches applied
Database optimized
SSL certificate valid (> 30 days)
Audit logs archived
Performance report generated
Disaster recovery drill passed

Summary: _____

11. ESCALATION CONTACTS

Issue	Contact	Phone	Email
Application down	DevOps Lead	+62-xxx	devops@kopkarskf.com
Database crash	DBA	+62-xxx	dba@kopkarskf.com
Security incident	Security Team	+62-xxx	security@kopkarskf.com
Critical bug	Dev Team	+62-xxx	dev@kopkarskf.com

Escalation Path: 1. IT Support (1st tier) - Basic troubleshooting 2. System Admin (2nd tier) - Server/database issues 3. DevOps Lead (3rd tier) - Complex infrastructure 4. CTO (4th tier) - Critical business impact

12. REFERENCES

- Laravel Documentation

- TROUBLESHOOTING.md
 - DEPLOYMENT.md
 - SECURITY.md
-

Maintainer: IT Operations Team

Contact: ops@kopkarskf.com

Last Reviewed: 17 January 2026

Next Review: 17 April 2026 (Quarterly)