Contents

SECURITY DOCUMENTATION

Document Information

Application: Koperasi Karyawan SKF
Version: 2.0.0
Security Level: Confidential
Last Updated: 17 January 2026
Next Review: 17 July 2026 (6 months)

Security Overview

Aplikasi Koperasi Karyawan SKF menerapkan **multi-layer security approach** untuk melindungi: - Data keuangan anggota (simpanan, pinjaman, SHU) - Data pribadi (NIK, alamat, foto, kontak) - Transaksi bisnis (POS, pembelian, expense) - Dokumen resmi koperasi

Security Standards Compliance: - OWASP Top 10 Protection - PCI-DSS Level 2 (Payment handling) - ISO 27001 Guidelines - UU No. 27 Tahun 2022 (Perlindungan Data Pribadi - Indonesia)

1. AUTHENTICATION & AUTHORIZATION

1.1 Authentication Mechanism

Primary Method: Session-based authentication (Laravel Sanctum)

Login Process:

```
User credentials (email + password)
    ↓
Validation
    ↓
Bcrypt hash comparison
    ↓
Session creation (120 min lifetime)
    ↓
CSRF token generation
    ↓
Access granted
```

Security Features: - Password hashing: **Bcrypt** (cost factor: 12) - Session timeout: **120 minutes** (configurable) - CSRF protection: **Enabled** on all forms - Rate limiting: **60 requests/minute** per IP - Login throttling: **5 failed attempts** = account locked (15 mins)

1.2 Password Policy

Requirements (Enforced): - Minimum length: **8 characters** - Must contain: - At least 1 uppercase letter - At least 1 lowercase letter - At least 1 number - Recommended: Include special characters (!@#$%^&*)

Password Reset: - Reset link valid for: **60 minutes** - Sent via: Encrypted email (TLS) - Token: Single-use, one-time only

Password Storage:

```php
// NEVER stored in plain text
// Hashed using Bcrypt with salt
password_hash($password, PASSWORD_BCRYPT, ['cost' => 12]);
```

Forbidden Practices: - Default/weak passwords (e.g., "password", "123456") - Password sharing between users - Storing passwords in browser without encryption - Sending passwords via unencrypted channels

1.3 Role-Based Access Control (RBAC)

Roles Hierarchy:

```
System Admin (Highest privilege)
      Admin
         Pengurus
            Manager Toko
               Kasir
                  Anggota (Lowest privilege)
```

Access Control Matrix:

Feature	Admin	Pengurus	Manager	Kasir	Anggota
Dashboard	Full	Full	Limited	Limited	View
Manage Members					
Approve Loans					
View All Loans					Own only
POS Operations					
Generate Reports			Limited		
SHU Calculation					
Settings/Config					
Backup/Restore					

Permission Enforcement: - **Middleware Level:** Route protection - **Controller Level:** Authorization checks - **View Level:** Conditional rendering - **Database Level:** Query scoping

Example Implementation:

```
// Middleware
Route::middleware(['auth', 'role:admin,pengurus'])->group(function() {
    Route::resource('members', MemberController::class);
});

// Controller
public function approve(Loan $loan) {
    $this->authorize('approve', $loan);
    // ...
}

// Policy
public function approve(User $user, Loan $loan) {
    return $user->role === 'admin' || $user->role === 'pengurus';
}
```

---

2. APPLICATION SECURITY

2.1 OWASP Top 10 Protection

A01: Broken Access Control PROTECTED

- All routes protected by authentication middleware
- Authorization checks on every sensitive operation
- No direct object reference without validation

A02: Cryptographic Failures PROTECTED

- HTTPS/TLS 1.3 enforced (production)
- Sensitive data encrypted at rest
- Password hashed with bcrypt
- Database connection encrypted

A03: Injection PROTECTED

- **SQL Injection:** Eloquent ORM (parameterized queries)
- **XSS:** Blade auto-escaping `{{ $var }}`
- **Command Injection:** No shell_exec/system calls with user input

Example:

```
//  SAFE - Eloquent ORM
User::where('email', $request->email)->first();

//  UNSAFE - Raw SQL (avoided)
DB::select("SELECT * FROM users WHERE email = '$email'");
```

A04: Insecure Design MITIGATED

- Security requirements defined upfront
- Threat modeling conducted
- Security reviews in design phase

A05: Security Misconfiguration PROTECTED

- `APP_DEBUG=false` in production
- Error messages sanitized (no stack traces to users)
- Unnecessary services disabled
- Default credentials changed

A06: Vulnerable Components MONITORED

- Dependencies updated regularly (`composer update`)
- Security advisories monitored
- Laravel framework kept up-to-date

Check for vulnerabilities:

```
composer audit
npm audit
```

A07: Identification & Authentication Failures PROTECTED

- Strong password policy enforced
- Multi-factor authentication (planned)
- Session management secure
- Credential stuffing prevention (rate limiting)

A08: Software & Data Integrity Failures   PROTECTED

- Code signing (Git commits)
- Dependency integrity (composer.lock)
- Auto-update disabled (manual review required)

A09: Security Logging & Monitoring   IMPLEMENTED

- All authentication events logged
- Failed login attempts tracked
- Audit trail for critical operations
- Log retention: 90 days

A10: Server-Side Request Forgery (SSRF)   PROTECTED

- URL validation before external requests
- Whitelist of allowed domains
- No user-controlled URLs in APIs

2.2 CSRF Protection

Enabled Globally: - All POST/PUT/PATCH/DELETE requests require CSRF token - Token included in all forms via `@csrf` directive - Token validated by `VerifyCsrfToken` middleware

Example:

```html
<form method="POST" action="/loans">
    @csrf
    <!-- CSRF token auto-included -->
    <input type="text" name="amount">
    <button type="submit">Submit</button>
</form>
```

AJAX Requests:

```javascript
axios.defaults.headers.common['X-CSRF-TOKEN'] =
    document.querySelector('meta[name="csrf-token"]').content;
```

2.3 XSS Prevention

Blade Auto-Escaping:

```
{{--  SAFE - Auto-escaped --}}
{{ $user->name }}

{{--  DANGEROUS - Raw output (only use if necessary) --}}
{!! $htmlContent !!}
```

Content Security Policy (CSP):

```
Content-Security-Policy:
    default-src 'self';
    script-src 'self' 'unsafe-inline' https://app.midtrans.com;
    img-src 'self' data: https:;
```

2.4 Input Validation

Server-Side Validation (Mandatory):

```
$request->validate([
    'email' => 'required|email|unique:users',
    'amount' => 'required|numeric|min:0|max:100000000',
    'nik' => 'required|digits:16',
    'phone' => 'required|regex:/^62[0-9]{9,12}$/',
]);
```

Client-Side Validation (UX): - HTML5 validation attributes - JavaScript validation (Alpine.js) - Real-time feedback

File Upload Validation:

```
$request->validate([
    'photo' => 'required|image|mimes:jpeg,png,jpg|max:2048', // 2MB max
    'receipt' => 'required|file|mimes:jpeg,png,pdf|max:5120', // 5MB max
]);
```

---

3. DATA SECURITY

3.1 Data Classification

Level	Examples	Protection
Critical	NIK, Password, PIN	Encrypted + Access restricted
Confidential	Simpanan, Pinjaman, Address	Access controlled + Audit logged
Internal	Product prices, Stock	Access controlled
Public	Announcements, AD/ART	No special protection

3.2 Data Encryption

Data at Rest: - Database: Transparent Data Encryption (TDE) - optional - Sensitive fields: Laravel encryption (`encrypt()` helper) - Backup files: GPG encrypted before storage

Data in Transit: - HTTPS/TLS 1.3 (production) - Certificate: Let's Encrypt (auto-renew) - HSTS enabled: `Strict-Transport-Security: max-age=31536000`

Encrypted Fields (Example):

```
// Automatic encryption
protected $casts = [
    'nik' => 'encrypted',
];

// Usage
$member->nik = '3201234567890123'; // Auto-encrypted on save
echo $member->nik; // Auto-decrypted on read
```

3.3 Data Retention & Deletion

Retention Policy: | **Data Type** | **Retention** | **Justification** | |—|—|—| | Transaction logs | 7 years | Tax/Audit requirement | | Audit logs | 3 years | Compliance | | Personal data (inactive member) | 1 year after resignation | GDPR-like | | Backup files | 90 days | Storage optimization | | Session data | 120 minutes | Security |

Right to be Forgotten: - Member dapat request penghapusan data - Data dianonimisasi (bukan dihapus total) untuk menjaga integritas audit trail - Proses approval diperlukan (Admin + Legal)

Data Anonymization:

```php
// Anonymize member data
$member->update([
    'name' => 'DELETED_USER_' . $member->id,
    'email' => 'deleted_' . $member->id . '@anonymized.local',
    'nik' => null,
    'phone' => null,
    'address' => null,
    'photo' => null,
]);
```

---

4. NETWORK SECURITY

4.1 Firewall Configuration

Server-Level (UFW):

```bash
# Allow only necessary ports
ufw allow 80/tcp    # HTTP (redirect to HTTPS)
ufw allow 443/tcp   # HTTPS
ufw allow 22/tcp    # SSH (limited to admin IPs)
ufw deny from any to any
ufw enable
```

Allowed IPs for SSH: - Admin IP 1: [Specify] - Admin IP 2: [Specify] - All others: Denied

---

4.2 DDoS Protection

Cloudflare (Recommended): - Enable "I'm Under Attack" mode if needed - Rate limiting: 100 req/10s - Challenge malicious bots

Application-Level:

```php
// Rate limiting middleware
Route::middleware('throttle:60,1')->group(function() {
    // Public routes
});

Route::middleware('throttle:login')->group(function() {
    Route::post('/login'); // 5 attempts per minute
});
```

4.3 API Security

Authentication: - API Token (Bearer token) - Token stored in `personal_access_tokens` table - Expires after: 30 days (configurable)

Example:

```
curl -X GET https://kopkarskf.com/api/members \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Accept: application/json"
```

Rate Limiting: - API: 60 requests/minute per token - Webhook: No limit (IP whitelist only)

---

5. MONITORING & AUDIT

5.1 Security Monitoring

Real-time Alerts: - Multiple failed login attempts (> 5 in 1 minute) - Unauthorized access attempts (403 errors) - Unusual data access patterns - Large data exports - Critical configuration changes

Alert Channels: - Email: admin@kopkarskf.com - WhatsApp: Security team group - Slack: #security-alerts (if configured)

5.2 Audit Logging

Events Logged: | **Category** | **Events** | |—|—| | Authentication | Login, logout, failed login, password reset | | Authorization | Permission denied (403) | | Data Changes | Create, update, delete (critical tables) | | Financial | Loan approval, SHU distribution, payment recording | | System | Backup, restore, configuration changes |

Audit Log Format:

```
{
    "id": 12345,
    "user_id": 5,
    "action": "update",
    "model": "Loan",
    "model_id": 789,
    "changes": {
        "status": ["pending", "approved"],
        "approved_by": [null, 5],
        "approved_at": [null, "2026-01-17 19:55:00"]
    },
    "ip_address": "192.168.1.100",
    "user_agent": "Mozilla/5.0...",
    "created_at": "2026-01-17 19:55:00"
}
```

Log Retention: - Storage: Database (`audit_logs` table) + File (`storage/logs/audit.log`) - Retention: 3 years - Archive: Yearly to cold storage

Access to Logs: - View: Admin only - Export: Admin only (with approval) - Tamper-proof: Write-only (no delete)

5.3 Security Metrics

KPIs to Track: | **Metric** | **Target** | **Alert If** | |—|—|—| | Failed login rate | < 5% | > 10% | | 403 errors/day | < 50 | > 100 | | Password reset requests/day | < 10 | > 20 | | Suspicious IP access | 0 | > 0 | | Unauthorized data access | 0 | > 0 | | Critical bugs unpatched | 0 | > 0 |

---

6. INCIDENT RESPONSE

6.1 Incident Classification

Severity	Definition	Response Time
P0 - Critical	Data breach, system compromise	< 1 hour
P1 - High	Unauthorized access, DoS attack	< 4 hours
P2 - Medium	Suspicious activity, minor breach	< 24 hours
P3 - Low	Security scan findings, policy violation	< 1 week

6.2 Incident Response Plan

Step 1: Identification - Detect anomaly via monitoring - Classify severity - Alert security team

Step 2: Containment - Isolate affected systems - Block malicious IPs - Revoke compromised credentials - Enable maintenance mode if needed

Step 3: Eradication - Identify root cause - Remove malware/backdoor - Patch vulnerability - Update firewall rules

Step 4: Recovery - Restore from clean backup - Verify system integrity - Resume normal operations - Monitor for recurrence

Step 5: Post-Incident - Document timeline & actions - Root cause analysis - Update security policies - Conduct lessons learned meeting

---

6.3 Emergency Contacts

Role	Name	Phone	Email
Security Lead	[TBD]	+62-xxx-xxxx-xxxx	security@kopkarskf.com
System Admin	[TBD]	+62-xxx-xxxx-xxxx	sysadmin@kopkarskf.com
Dev Lead	[TBD]	+62-xxx-xxxx-xxxx	dev@kopkarskf.com
Business Owner	[Ketua]	+62-xxx-xxxx-xxxx	ketua@kopkarskf.com

---

7. BACKUP & DISASTER RECOVERY

7.1 Backup Strategy

Schedule: - **Daily:** Full database backup (02:00 WIB) - **Weekly:** Full system backup (Sunday 03:00 WIB) - **Monthly:** Archive to cold storage

Backup Locations: - **Primary:** Local server (`storage/backups/`) - **Secondary:** Google Drive (encrypted) - **Tertiary:** External HDD (monthly, offline storage)

Encryption:

```
# Backup encrypted with GPG
gpg --encrypt --recipient admin@kopkarskf.com backup.sql
```

Verification: - Daily: Automated integrity check - Monthly: Test restore to staging environment

---

7.2 Disaster Recovery

RTO (Recovery Time Objective): < 4 hours
RPO (Recovery Point Objective): < 24 hours

Disaster Scenarios & Response:

Scenario	Impact	Recovery Steps
Database corruption	High	Restore from last backup, replay transaction logs
Server hardware failure	Critical	Migrate to backup server, restore data
Ransomware attack	Critical	Isolate, wipe, restore from clean backup
Accidental data deletion	Medium	Restore specific tables from backup
DDoS attack	Medium	Enable Cloudflare protection, scale resources

Failover Plan: 1. Activate disaster recovery server 2. Update DNS to point to backup 3. Restore latest backup 4. Verify data integrity 5. Resume operations 6. Notify stakeholders

---

8. COMPLIANCE & POLICIES

8.1 Privacy Policy

Data Collection: - What: Name, NIK, email, phone, address, photo, financial data - Why: Membership management, transaction processing - How: User registration, admin input, POS transactions - Retention: As per retention policy

Data Sharing: - Internal: Only with authorized personnel - External: Payment gateway (Midtrans) - encrypted - Third-party: NEVER sold or shared

User Rights: - Right to access personal data - Right to correction - Right to deletion (anonymization) - Right to data portability

---

8.2 Acceptable Use Policy

Permitted: - Access for legitimate business purposes - Personal data viewing (own data only) - Reporting bugs/issues

Prohibited: - Sharing login credentials - Accessing others' data without authorization - Attempting to bypass security controls - Data scraping/harvesting - Using application for illegal activities

Violations: - First offense: Warning - Second offense: Account suspension - Third offense: Account termination + legal action

---

8.3 Third-Party Security

Midtrans (Payment Gateway): - PCI-DSS Level 1 certified - Tokenization for card data - 3D Secure authentication - API credentials stored in `.env` (not version controlled)

Google Drive (Backup): - OAuth 2.0 authentication - Service account with limited scope - Encrypted files only

Email Provider (SMTP): - TLS encryption required - App-specific password (not main password)

---

9. SECURITY TESTING

9.1 Penetration Testing

Frequency: Annually (or after major release)

Scope: - Authentication & authorization - Input validation - SQL injection, XSS, CSRF - API security - Session management - File upload vulnerabilities

Tools: - OWASP ZAP - Burp Suite - Nikto - SQLMap

Report: - Findings documented - Severity classification - Remediation recommendations - Re-test after fixes

---

9.2 Code Review

Security Code Review Checklist: - [ ] No hardcoded credentials - [ ] Input validation on all user inputs - [ ] Authorization checks on sensitive operations - [ ] Parameterized queries (no raw SQL) - [ ] Files uploaded validated (type, size) - [ ] Sensitive data encrypted - [ ] Error messages sanitized - [ ] Audit logging for critical actions

---

10. SECURITY TRAINING

10.1 User Security Awareness

Topics: - Password best practices - Phishing awareness - Social engineering - Safe browsing - Incident reporting

Frequency: Annually for all users

10.2 Developer Security Training

Topics: - OWASP Top 10 - Secure coding practices - Laravel security features - Dependency management - Security testing

Frequency: Quarterly

SECURITY CONTACT

Report Security Issues: - Email: security@kopkarskf.com - Emergency Hotline: +62-xxx-xxxx-xxxx - Bug Bounty: [If implemented]

PGP Key: [Public key for encrypted communication]

Response SLA: - Critical: < 1 hour - High: < 4 hours - Medium: < 24 hours - Low: < 1 week

---

Document Owner: Security Team
Approved By: [CTO / Security Lead]
Next Review: 17 July 2026

Version History: - v1.0 (17 Jan 2026) - Initial document