🔍 ANALISIS KODE & SETUP PRODUCTION - AVEVA-PI

📁 Struktur Repository (Jaringan Privat)

```
AVEVA-PI/
├── avevapi/          # Backend API (Node.js + SQLite)
├── frontend/         # Frontend App (Next.js + React)
└── wa/              # WhatsApp Bot (Node.js + whatsapp-web.js)
```

🔧 1. BACKEND (avevapi/) - Node.js + Express + SQLite

📦 Dependencies yang Perlu Install:

```
# Production dependencies (wajib)
npm install axios bcrypt better-sqlite3 cookie-parser cors dotenv express express-rate-limit express-validator fetch-cookie helmet joi jsonwebtoken mysql2 node-fetch oracledb pg tough-cookie uuid

# Dev dependencies (opsional untuk production)
npm install --save-dev eslint nodemon
```

⚙️ Environment Variables (.env):

```
# Server Configuration
PORT=8001
HOST=0.0.0.0
NODE_ENV=production

# API Configuration
API_BASE_URL=http://localhost:8001/api
API_KEY=universal-api-key-2025

# Database (SQLite - default)
```

```
DATABASE_URL=./data/app.db

# JWT Authentication
JWT_SECRET=your-super-secure-jwt-secret-here-minimum-32-chars

# CORS (untuk frontend access)
CORS_ORIGIN=http://localhost:3000,http://your-server-ip:3000

# WhatsApp Integration
WA_TIMEOUT=30000

# Security
FORCE_HTTPS=false
ALLOW_NGROK_COOKIES=false
```

🚀 Cara Setup di Server:

```
# 1. Clone repository
git clone https://your-private-git-server/avevapi.git
cd avevapi

# 2. Install dependencies
npm install --production

# 3. Setup environment
cp .env.example .env  # Edit sesuai kebutuhan
nano .env  # Configure environment variables

# 4. Create data directory
mkdir -p data

# 5. Test run
npm start

# 6. Production run (recommended with PM2)
npm install -g pm2
pm2 start main.js --name "avevapi-backend"
pm2 save
pm2 startup
```

📋 Port & Network:

- **Port**: 8001 (default)
- **Protocol**: HTTP (internal network)
- **Database**: SQLite file-based (`./data/app.db`)

🎨 2. FRONTEND (frontend/) - Next.js + React + TypeScript

📦 Dependencies yang Perlu Install:

```
# Production dependencies
npm install react react-dom next

# Dev dependencies (build time only)
npm install --save-dev typescript @types/node @types/react @types/react-dom
tailwindcss eslint eslint-config-next
```

⚙️ Environment Variables (.env.local):

```
# API Configuration
NEXT_PUBLIC_API_URL=http://your-server-ip:8001
NEXT_PUBLIC_WS_URL=ws://your-server-ip:8001

# App Configuration
NEXT_PUBLIC_APP_NAME=AVEVA-PI
NEXT_PUBLIC_APP_VERSION=1.0.0
```

🚀 Cara Setup di Server:

```
# 1. Clone repository
git clone https://your-private-git-server/frontend.git
cd frontend

# 2. Install dependencies
npm install --production

# 3. Setup environment
cp .env.local.example .env.local  # Jika ada
nano .env.local  # Configure API URLs

# 4. Build untuk production
npm run build

# 5. Test production build
npm start
```

```
# 6. Production run dengan PM2
pm2 start npm --name "avevapi-frontend" -- start
pm2 save
```

📋 Port & Network:

- **Development**: Port 3000
- **Production**: Port 3000 (default Next.js)
- **Build Output**: `.next/` directory
- **Static Files**: Served by Next.js

🤖 3. WHATSAPP BOT (wa/) - Node.js + whatsapp-web.js

📦 Dependencies yang Perlu Install:

```
# Production dependencies (wajib)
npm install axios dotenv form-data puppeteer qrcode qrcode-terminal whatsapp-web.js

# Dev dependencies (opsional)
npm install --save-dev nodemon
```

⚙️ Environment Variables (.env):

```
# API Configuration
API_BASE_URL=http://localhost:8001
API_KEY=universal-api-key-2025

# WhatsApp Configuration
SESSION_PATH=./sessions
WA_TIMEOUT=30000

# Bot Settings
NODE_ENV=production
PORT=8002

# Puppeteer (untuk WhatsApp Web automation)
```

```
PUPPETEER_SKIP_CHROMIUM_DOWNLOAD=false
PUPPETEER_EXECUTABLE_PATH=/usr/bin/chromium-browser
```

🚀 Cara Setup di Server:

```
# 1. Clone repository
git clone https://your-private-git-server/wa.git
cd wa

# 2. Install dependencies
npm install --production

# 3. Install system dependencies (untuk Puppeteer)
sudo apt-get update
sudo apt-get install -y chromium-browser

# 4. Setup environment
cp .env.example .env  # Jika ada
nano .env  # Configure API URLs

# 5. Create required directories
mkdir -p sessions downloads failed-messages .status

# 6. Test run (akan generate QR code)
npm start

# 7. Production run dengan PM2
pm2 start index.js --name "avevapi-whatsapp-bot"
pm2 save
```

📋 Port & Network:

- **Port**: 8002 (untuk health check)
- **WhatsApp Web**: Auto-managed by puppeteer
- **Session Storage**: `./sessions/` directory
- **QR Code**: Generated on first run

🖥️ SERVER REQUIREMENTS (Minimal)

Hardware:

- **RAM**: 2GB minimum, 4GB recommended
- **CPU**: 1 core minimum, 2 cores recommended
- **Storage**: 10GB minimum (untuk logs, sessions, database)

Software:

```
# Ubuntu/Debian Server
sudo apt update
sudo apt install -y nodejs npm chromium-browser git curl

# Verify versions
node --version  # Should be 18+
npm --version   # Should be 8+
chromium-browser --version  # Should be installed
```

🚀 PRODUCTION DEPLOYMENT STEPS

Step 1: Server Preparation

```
# Update system
sudo apt update && sudo apt upgrade -y

# Install Node.js 18+ (via NodeSource)
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-get install -y nodejs

# Install PM2 globally
sudo npm install -g pm2

# Install Chromium (untuk WhatsApp bot)
sudo apt-get install -y chromium-browser

# Create app directory
sudo mkdir -p /opt/aveva-pi
sudo chown $USER:$USER /opt/aveva-pi
cd /opt/aveva-pi
```

Step 2: Clone & Setup Backend

```
# Clone backend
git clone https://your-private-git-server/avevapi.git backend
cd backend

# Install & configure
npm install --production
cp .env.example .env
nano .env  # Edit configuration

# Create data directory
mkdir -p data

# Start backend
pm2 start main.js --name "avevapi-backend"
```

Step 3: Clone & Setup Frontend

```
# Clone frontend
cd /opt/aveva-pi
git clone https://your-private-git-server/frontend.git frontend
cd frontend

# Install & build
npm install --production
cp .env.local.example .env.local
nano .env.local  # Configure API URL
npm run build

# Start frontend
pm2 start npm --name "avevapi-frontend" -- start
```

Step 4: Clone & Setup WhatsApp Bot

```
# Clone bot
cd /opt/aveva-pi
git clone https://your-private-git-server/wa.git wa
cd wa

# Install & configure
npm install --production
cp .env.example .env
nano .env  # Configure API URL
mkdir -p sessions downloads failed-messages .status
```

```
# Start bot
pm2 start index.js --name "avevapi-whatsapp-bot"
```

Step 5: Configure PM2 Startup

```
# Save PM2 configuration
pm2 save

# Generate startup script
pm2 startup

# Follow the instructions to enable auto-start
sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u
$USER --hp $HOME
```

Step 6: Nginx Reverse Proxy (Optional)

```
# Install Nginx
sudo apt install nginx

# Create configuration
sudo nano /etc/nginx/sites-available/aveva-pi

# Add this configuration:
server {
    listen 80;
    server_name your-server-ip;

    # Frontend
    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    # Backend API
    location /api {
        proxy_pass http://localhost:8001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
    }
}
```

```
# Enable site
sudo ln -s /etc/nginx/sites-available/aveva-pi /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

📊 MONITORING & MAINTENANCE

PM2 Management:

```
# Check status
pm2 status

# View logs
pm2 logs avevapi-backend
pm2 logs avevapi-frontend
pm2 logs avevapi-whatsapp-bot

# Restart services
pm2 restart all

# Update deployments
cd /opt/aveva-pi/backend && git pull && npm install && pm2 restart avevapi-backend
cd /opt/aveva-pi/frontend && git pull && npm install && npm run build && pm2
restart avevapi-frontend
cd /opt/aveva-pi/wa && git pull && npm install && pm2 restart avevapi-whatsapp-bot
```

Backup Strategy:

```
# Database backup (SQLite)
cp /opt/aveva-pi/backend/data/app.db /opt/aveva-pi/backup/app-$(date +%Y%m%d).db

# WhatsApp sessions backup
tar -czf /opt/aveva-pi/backup/sessions-$(date +%Y%m%d).tar.gz /opt/aveva-
pi/wa/sessions/
```

⚠️ IMPORTANT NOTES

1. **Environment Variables**: Pastikan semua `.env` files dikonfigurasi dengan benar
2. **Firewall**: Buka port 80, 3000, 8001, 8002 sesuai kebutuhan
3. **SSL**: Untuk production, gunakan HTTPS dengan sertifikat valid
4. **Database**: SQLite cocok untuk internal, tapi consider PostgreSQL untuk high-traffic
5. **Security**: Jangan commit `.env` files ke git
6. **Updates**: Test di staging environment dulu sebelum production

---

🔍 TROUBLESHOOTING

Backend tidak start:

```
cd /opt/aveva-pi/backend
npm start  # Check error messages
tail -f ~/.pm2/logs/avevapi-backend-out.log
```

Frontend blank page:

```
cd /opt/aveva-pi/frontend
cat .env.local  # Check API URL configuration
curl http://localhost:8001/api/health  # Test backend connectivity
```

WhatsApp bot QR code tidak muncul:

```
cd /opt/aveva-pi/wa
ls -la sessions/  # Check session directory
chromium-browser --version  # Check Chromium installation
```

Setup ini untuk jaringan privat/internal server. Apakah ada spesifikasi khusus server atau konfigurasi yang berbeda?