🚀 IDRT Subgraph Deployment Guide

📁 **Repository:** https://github.com/yeheskieltame/Rupiah-tracking

Step-by-Step Deployment ke The Graph Studio

0. **Clone Repository (Optional)**

Jika ingin clone repository ini:

```
git clone https://github.com/yeheskieltame/Rupiah-tracking.git
cd Rupiah-tracking
npm install
```

1. **Setup Account & Subgraph**

1. **Buat akun di The Graph Studio**
2. **Connect wallet** (MetaMask, WalletConnect, etc.)
3. **Create new subgraph:**
    - Name: `idrt-transaction-analyzer`
    - Slug: `idrt-transaction-analyzer`
    - Subtitle: "IDRT Token Transaction Analysis"
    - Description: "Comprehensive IDRT transaction tracking and wallet analytics"

2. **Install Dependencies**

```
# Install Graph CLI globally
npm install -g @graphprotocol/graph-cli

# Verify installation
graph --version
```

3. **Authentication**

Copy deploy key dari subgraph studio, kemudian:

```
# Authenticate dengan deploy key
graph auth --studio YOUR_DEPLOY_KEY_HERE
```

4. **Deploy Subgraph**

```
# Clone repository (if not already)
git clone https://github.com/yeheskieltame/Rupiah-tracking.git
cd Rupiah-tracking

# Install dependencies
npm install

# Generate types
npm run codegen

# Final build
npm run build

# Deploy ke studio
graph deploy --studio idrt-transaction-analyzer
```

5. **Deployment Output**

Jika berhasil, Anda akan melihat:

```
✔ Upload subgraph to IPFS

Build completed: QmHash...

Deployed to https://thegraph.com/studio/subgraph/idrt-transaction-
analyzer

Subgraph endpoints:
Queries (HTTP):     https://api.studio.thegraph.com/query/YOUR_ID/idrt-
transaction-analyzer/v0.0.1
```

6. **Publish ke Decentralized Network (Optional)**

```
# Publish ke mainnet (memerlukan GRT untuk curation)
graph publish --studio idrt-transaction-analyzer
```

📊 Testing Your Deployment

Test Query di Studio

Buka subgraph di studio dan test query ini:

```
{
  transfers(first: 5, orderBy: blockTimestamp, orderDirection: desc) {
    id
```

```
    from
    to
    value
    blockTimestamp
    transactionHash
  }
}
```

Test dengan JavaScript

```
const SUBGRAPH_URL =
'https://api.studio.thegraph.com/query/YOUR_ID/idrt-transaction-
analyzer/v0.0.1';

async function testQuery() {
  const query = `
    {
      transfers(first: 5) {
        from
        to
        value
        blockTimestamp
      }
    }
  `;

  const response = await fetch(SUBGRAPH_URL, {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ query })
  });

  const data = await response.json();
  console.log(data);
}

testQuery();
```

🔍 Verification Steps

1. **Check Sync Status**

- Buka subgraph di studio
- Check "Indexing Status" - harus "Synced"
- Verify latest block number

2. **Verify Data Quality**

```
# Check total transfers
{
  globalStat(id: "global") {
    totalTransfers
    totalVolume
  }
}

# Check recent activity
{
  transfers(first: 1, orderBy: blockTimestamp, orderDirection: desc) {
    blockTimestamp
    transactionHash
  }
}
```

3. **Test Wallet Queries**

```
# Test dengan wallet address yang aktif
{
  account(id: "0x998ffe1e43facffb941dc337dd0468d52ba5b48a") {
    transferCount
    totalSent
    totalReceived
  }
}
```

🛠️ Troubleshooting

Common Issues

1. Sync Issues:

```
# Check logs di studio
# Restart indexing jika diperlukan
```

2. Query Errors:

- Pastikan field names sesuai schema
- Check GraphQL syntax
- Verify entity relationships

3. Missing Data:

- Check start block di subgraph.yaml
- Verify contract address

- Check ABI compatibility

Performance Tips

1. Efficient Queries:

```
# Good - specific fields
{
  transfers(first: 100) {
    from
    to
    value
  }
}

# Avoid - too many fields at once
{
  transfers(first: 1000) {
    # all fields...
  }
}
```

2. Pagination:

```
# Use skip for pagination
{
  transfers(first: 100, skip: 200) {
    from
    to
    value
  }
}
```

📈 Production Monitoring

Health Checks

```
// Monitor subgraph health
async function checkHealth() {
  const query = `
    {
      _meta {
        block {
          number
          timestamp
        }
        hasIndexingErrors
```

```
        }
      }
    `;
    // Implementation...
}
```

Alert Setup

Monitor for:

- Sync lag > 100 blocks
- Query error rate > 5%
- Index errors
- Performance degradation

🔄 Updates & Versioning

Deploy New Version

```
# Make changes to schema/mappings
# Update version in package.json

npm run codegen
npm run build
graph deploy --studio idrt-transaction-analyzer --version-label v0.0.2
```

Migration Strategy

1. Deploy new version
2. Test thoroughly
3. Update frontend endpoints
4. Monitor both versions
5. Deprecate old version

📞 Support

- **The Graph Discord**: discord.gg/graphprotocol
- **Documentation**: thegraph.com/docs
- **Forum**: forum.thegraph.com

Your IDRT analysis subgraph is ready for production! 🎉