

**T.C.**  
**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ**  
**YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ**



**PYTHON İLE EKSTÜRÜZYON PRES SİMÜLASYONU VE GRAFANA İLE  
MONİTORİNG**

**ENES ÖZTÜRK**

**DANIŞMAN**  
**Dr. HÜSEYİN PARMAKSIZ**

**YBS464 Bitirme Çalışması**

**BİLECİK 2025**

## **BİLDİRİM**

Bu çalışmada bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

## **DECLARATION**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

**İmza**

**ENES ÖZTÜRK**

**Tarih: 24 Mayıs 2025**

## Özet

Bu projede, alüminyum ekstrüzyon presinde oluşan üretim verilerinin Python ile simüle edilmesi, bu verilerin TimescaleDB tabanlı zaman serisi veritabanında saklanması ve Grafana arayüzü ile gerçek zamanlı olarak izlenmesi amaçlanmıştır. Endüstride yaygın olarak kullanılan PostgreSQL üzerine kurulu TimescaleDB eklentisi sayesinde yüksek hacimli ve zaman damgalı veriler performanslı şekilde depolanabilmiş ve analiz edilebilmiştir. Docker teknolojisi ile tüm altyapı modüller ve taşınabilir olarak tasarlanmıştır.

**Anahtar Kelimeler:** TimescaleDB, Docker, Python, Zaman Serisi Veritabanı, Grafana, Alüminyum Ekstrüzyon

## Abstract

In this project, it is aimed to simulate the production data of an aluminum extrusion press with Python, to store this data in a TimescaleDB-based time-series database, and to monitor it in real time via the Grafana interface. Thanks to TimescaleDB, which is a popular extension for PostgreSQL, high-volume and time-stamped data have been efficiently stored and analyzed. The whole infrastructure is designed as modular and portable using Docker technology.

**Keywords:** TimescaleDB, Docker, Python, Time-Series Database, Grafana, Aluminum Extrusion

## **ÖNSÖZ**

Bitirme çalışmamın başından sonuna kadar emeği geçen ve beni bu konuya yönlendiren saygı değer hocam ve danışmanım Sayın Ünvan AD-SOYAD’a tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

**İmza**

**ENES ÖZTÜRK**

**Tarih: 24 Mayıs 2025**

# İÇİNDEKİLER

<b>ÖNSÖZ</b>	<b>iv</b>
<b>İÇİNDEKİLER</b>	<b>v</b>
<b>ŞEKİLLER DİZİNİ</b>	<b>vi</b>
<b>1 GİRİŞ</b>	<b>1</b>
<b>2 YAZILIM MİMARİSİ VE YÖNTEM</b>	<b>2</b>
2.1 Kullanılan Teknolojiler . . . . .	2
2.2 Sistem Mimarisi . . . . .	2
2.3 TimescaleDB Avantajları . . . . .	2
2.4 Veri Tabanı Tasarımı . . . . .	3
<b>3 SİSTEMİN UYGULANMASI</b>	<b>4</b>
3.1 Veri Simülasyonu . . . . .	4
3.2 Python ile TimescaleDB'ye Bağlantı . . . . .	4
3.3 Docker Compose ile Kurulum . . . . .	4
3.4 Grafana Panelleri . . . . .	5
<b>4 BULGULAR VE TARTIŞMA</b>	<b>6</b>
<b>5 SONUÇ VE ÖNERİLER</b>	<b>7</b>
<b>6 EKLER</b>	<b>8</b>
6.1 Grafana Paneli Ekran Görüntüsü . . . . .	8
6.2 Sistem Mimari Diyagramı . . . . .	8

## ŞEKİLLER DİZİNİ

Şekil 2.1	Sistemin genel mimarisi: Simüle edilen veriler Python ile üretilip TimescaleDB’ye kaydedilir, Grafana ile izlenir. Tüm yapı Docker ile yönetilmektedir.	3
Şekil 6.1	Proje kapsamında kurulan Grafana panelinden bir ekran görüntüsü. . . . .	8
Şekil 6.2	Projenin Docker tabanlı sistem mimarisi diyagramı. . . . .	9

# 1 GİRİŞ

Endüstriyel üretim süreçlerinde, makinelerden alınan verilerin anlık olarak toplanması ve analiz edilmesi, verimlilik, kalite ve bakım yönetimi için kritik öneme sahiptir. Özellikle sürekli çalışan hatlarda, üretim ve proses parametrelerinin dijital ortamda izlenmesi hem işletme maliyetlerini düşürmekte hem de arıza ve duruşların önüne geçmektedir.

Alüminyum ekstrüzyon presleri de büyük hacimli, hızlı ve zaman damgalı veri üretir. Ancak, çoğu fabrikada doğrudan PLC verisine erişim sınırlı olabilmekte, test veya geliştirme amaçlı simülasyon verisine ihtiyaç duyulmaktadır. Bu projede, ekstrüzyon presi üretim verileri Python ile simüle edilmiş, zaman serisi veri yönetimi için PostgreSQL tabanlı TimescaleDB kullanılmış ve tüm sistem Docker konteynerlerinde koordine edilmiştir. TimescaleDB, büyük hacimli zaman serisi verileri çok daha hızlı sorgulama ve saklama imkanı sağladığından, endüstriyel monitoring projelerinde klasik ilişkisel veritabanlarından ayrılmaktadır.

Proje kapsamında geliştirilen sistem ile; sentetik (simüle) veriler gerçek bir üretim hattı gibi TimescaleDB'ye yazılmış ve Grafana arayüzünde gerçek zamanlı izlenmiştir.

## 2 YAZILIM MİMARİSİ VE YÖNTEM

Bu projede; Python, Docker, PostgreSQL ve TimescaleDB ile Grafana yazılımları entegre edilmiştir. Sistemin genel akışı ve yazılım mimarisi aşağıda detaylandırılmıştır.

### 2.1 Kullanılan Teknolojiler

- **Python:** Üretim verilerinin rastgele fakat gerçekçi aralıklarla simüle edilmesi için kullanılmıştır. Python scripti, belirli periyotlarda üretim, sıcaklık, basınç, arıza gibi verileri TimescaleDB'ye kaydeder.
- **PostgreSQL + TimescaleDB:** Temelde bir ilişkisel veritabanı olan PostgreSQL, TimescaleDB eklentisi ile zaman serisi veriler için optimize edilmiştir. Özellikle hypertable yapısı sayesinde milyonlarca satır verinin hızlı bir şekilde depolanması ve sorgulanması sağlanır.
- **Docker:** Tüm servisler (simülatör, veritabanı, monitoring arayüzü) izole konteynerlerde çalıştırılmış ve Docker Compose ile orkestrasyon sağlanmıştır.
- **Grafana:** TimescaleDB'den verileri çekip anlık olarak panellerde ve grafiklerde görselleştirmek için kullanılmıştır.

### 2.2 Sistem Mimarisi

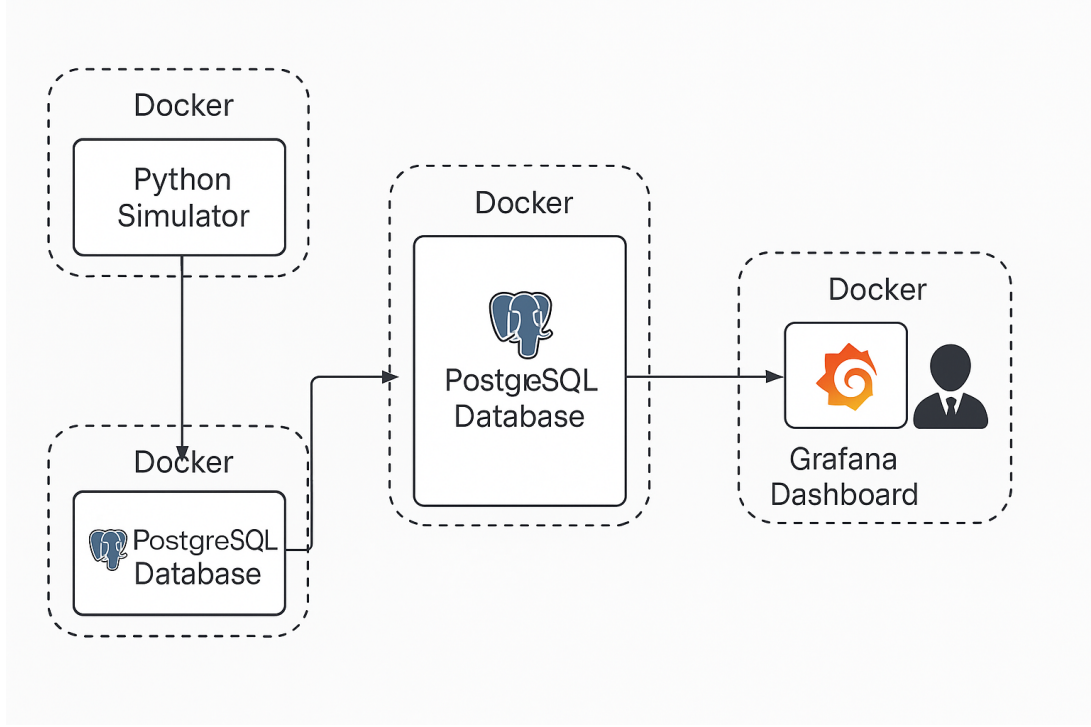
Sistem 4 ana bileşenden oluşmaktadır:

1. **Python Simülatörü:** Sürekli veri üretir ve TimescaleDB'ye yazar.
2. **TimescaleDB (PostgreSQL üzerinde):** Zaman damgalı tüm verileri hypertable olarak saklar.
3. **Grafana Dashboard:** TimescaleDB'ye bağlanıp verileri kullanıcıya gösterir.
4. **Kullanıcı/Operatör:** Grafana üzerinden süreci canlı izler.

### 2.3 TimescaleDB Avantajları

TimescaleDB, verileri otomatik bölümlere (chunk) ayırarak depolama ve sorgulamada performansı artırır. Ayrıca continuous aggregate ve veri arşivleme özellikleri sayesinde endüstriyel büyük veri uygulamaları için idealdir.





Şekil 2.1: Sistemin genel mimarisi: Simüle edilen veriler Python ile üretilip TimescaleDB'ye kaydedilir, Grafana ile izlenir. Tüm yapı Docker ile yönetilmektedir.

## 2.4 Veri Tabanı Tasarımı

Ana tablo, aşağıdaki gibi hypertable olarak tanımlanmıştır:

```
CREATE EXTENSION IF NOT EXISTS timescaledb;
CREATE TABLE uretim_verileri (
    id SERIAL PRIMARY KEY,
    timestamp TIMESTAMPTZ NOT NULL,
    uretim_miktari DOUBLE PRECISION,
    sicaklik DOUBLE PRECISION,
    basinc DOUBLE PRECISION,
    ariza_durumu INTEGER
);
SELECT create_hypertable('uretim_verileri', 'timestamp');
```

## 3 SİSTEMİN UYGULANMASI

### 3.1 Veri Simülasyonu

Python scripti ile, gerçek bir ekstrüzyon presinden beklenen parametreler simüle edilmiştir:

- Üretim Miktarı (800–1200 kg/saat)
- Sıcaklık (450–480 °C)
- Basınç (180–220 bar)
- Arıza Durumu (rastgele tetiklenir)
- Sipariş Durumu (aktif, tamamlanan vb.)

Her veri kaydı, zaman damgası ile birlikte TimescaleDB'ye yazılmıştır.

### 3.2 Python ile TimescaleDB'ye Bağlantı

```
import random, time, psycopg2
conn = psycopg2.connect(host="timescaledb", database="presdb", user="user")
cur = conn.cursor()
while True:
    uretim = random.uniform(800, 1200)
    sicaklik = random.uniform(450, 480)
    basinc = random.uniform(180, 220)
    ariza = random.choice([0, 1]) if random.random() < 0.1 else 0
    cur.execute("INSERT INTO uretim_verileri (timestamp, uretim_miktari,
    conn.commit()
    time.sleep(5)
```

### 3.3 Docker Compose ile Kurulum

```
version: '3'
services:
    timescaledb:
        image: timescale/timescaledb:latest-pg14
```

```
environment:
  POSTGRES_DB: presdb
  POSTGRES_USER: user
  POSTGRES_PASSWORD: pass
ports:
  - "5432:5432"
grafana:
  image: grafana/grafana
  ports:
    - "3000:3000"
simulator:
  build: ./simulator
  depends_on:
    - timescaledb
```

### 3.4 Grafana Panelleri

Kurulan paneller şunlardır:

- Üretim miktarı zaman grafiği
- Sıcaklık ve basınç trendleri
- Arıza sıklığı ve kayıtları
- Sipariş ve log tablosu

Tüm paneller TimescaleDB'den canlı veri çekmektedir.

## 4 BULGULAR VE TARTIŞMA

Kurulan sistem test edildiğinde, Python simülatörünün ürettiği veriler TimescaleDB üzerinde hızlıca saklanmış ve Grafana panellerinde anlık olarak görselleştirilmiştir. Özellikle TimescaleDB'nin hypertable mimarisi sayesinde yüksek miktarda veride bile sorgu performansı tatmin edici düzeyde kalmıştır.

Sistemin avantajları şunlardır:

- Zaman serisi veri için optimize mimari ile büyük veri miktarlarında yüksek performans.
- Endüstriyel monitoring uygulamaları için açık kaynaklı, ekonomik ve modüler bir çözüm.
- Docker altyapısı ile kolay kurulum ve bakım.
- Grafana ile modern, kullanıcı dostu bir izleme arayüzü.

Ayrıca, sistem gerçek üretim ortamına aktarıldığında gerçek zamanlı arıza takibi, kestirimci bakım ve süreç optimizasyonu için temel oluşturabilir.

## 5 SONUÇ VE ÖNERİLER

Bu projede, alüminyum ekstrüzyon presi için üretim verilerinin simülasyonu ve gerçek zamanlı izlenmesini sağlayan modern bir dijital altyapı başarıyla geliştirilmiştir. Python ile yazılan simülör, PostgreSQL veritabanı ve Grafana görselleştirme arayüzü ile endüstride kolayca uyarlanabilir ve genişletilebilir bir model sunulmuştur.

Sistem sayesinde:

- Üretim verilerinin, proses parametrelerinin ve olay günlüklerinin merkezi şekilde toplanıp yönetilmesi mümkün olmuştur.
- Gerçek zamanlı izleme ile süreç yönetiminde hızlı karar alma ve sorunların tespiti kolaylaşmıştır.
- Docker altyapısı ile taşınabilir, modüler ve hızlı kurulabilen bir mimari elde edilmiştir.

Gelecekteki çalışmalar için şu öneriler getirilebilir:

- Sisteme gerçek PLC bağlantısı ve IoT sensör entegrasyonu eklenerek doğrudan saha verileriyle çalışma yapılabilir.
- Otomatik uyarı/alarm sistemleri, e-posta bildirimleri ve rapor modülleri ile sistem daha da geliştirilebilir.
- Toplanan büyük verinin analiziyle kestirimci bakım (predictive maintenance) ve makine öğrenmesi tabanlı süreç iyileştirmeleri uygulanabilir.

Bu proje ile elde edilen bulgular, dijitalleşen üretim süreçlerinde açık kaynak teknolojilerin pratik ve ekonomik çözümler sunduğunu göstermektedir.

## 6 EKLER

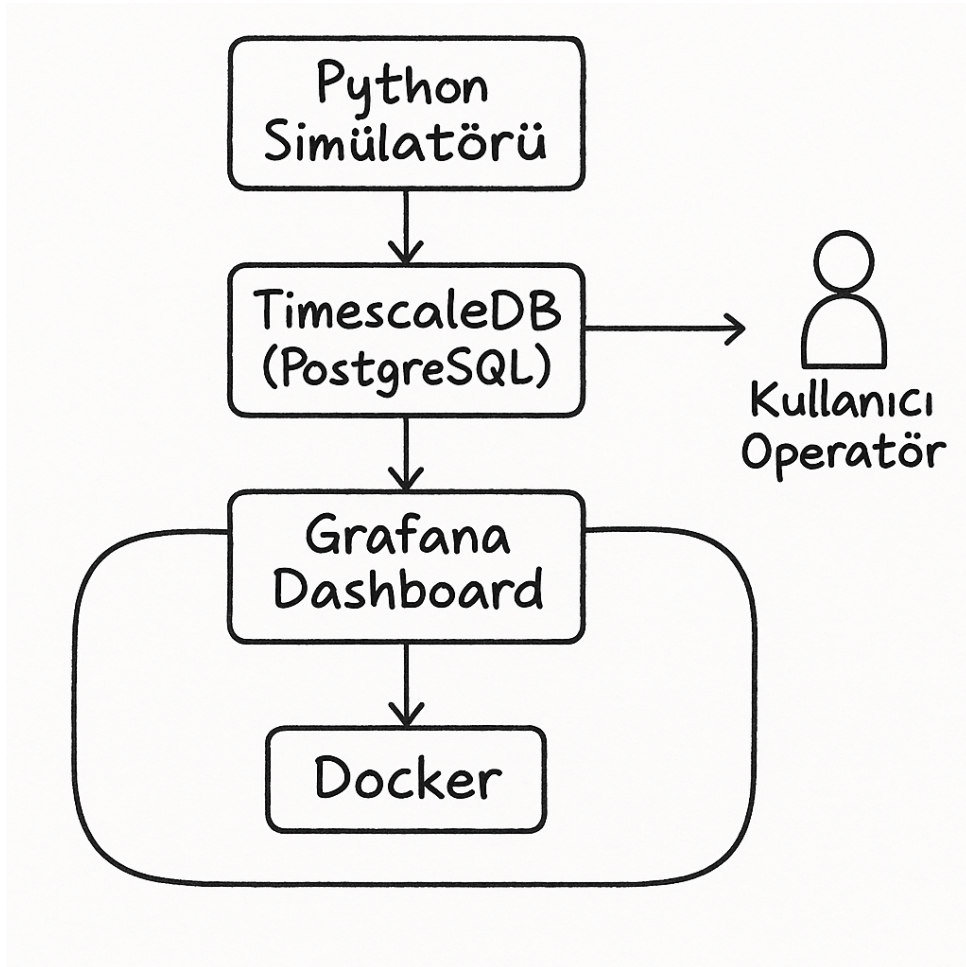
### 6.1 Grafana Paneli Ekran Görüntüsü

Aşağıda, projede oluşturulan Grafana izleme panelinin bir örnek ekran görüntüsü verilmiştir. Bu panelde net üretimi canlı bir şekilde izleyebilir ayrıca olay günlüğünden bütün olayları takip edebilme şansına da erişebiliriz.



Şekil 6.1: Proje kapsamında kurulan Grafana panelinden bir ekran görüntüsü.

### 6.2 Sistem Mimari Diyagramı



Şekil 6.2: Projenin Docker tabanlı sistem mimarisi diyagramı.

## **Kaynaklar**

- [1] Docker Documentation. <https://docs.docker.com/>
- [2] Grafana Documentation. <https://grafana.com/docs/>
- [3] PostgreSQL Documentation. <https://www.postgresql.org/docs/>
- [4] Python Documentation. <https://docs.python.org/3/>