

Servis Uygulamaları İçin Kural Tabanlı Proses Takip Yazılımının Geliştirilmesi

Cengiz SERTKAYA^{1*}, Fidan Khalilbayli²

¹Bilgisayar Mühendisliği Bölümü / Mühendislik Fakültesi, İstanbul Topkapı Üniversitesi, Türkiye

²BT Uygulama Çözümleri Bölümü, Eminevim, Türkiye

*(cengizsertkaya@topkapi.edu.tr)

(Received: 16 July 2024, Accepted: 24 July 2024)

(4th International Conference on Scientific and Academic Research ICSAR 2024, July 19 - 20, 2024)

ATIF/REFERENCE: Sertkaya, C. & Khalilbayli, F. (2024). Servis Uygulamaları İçin Kural Tabanlı Proses Takip Yazılımının Geliştirilmesi. *International Journal of Advanced Natural Sciences and Engineering Researches*, 8(6), 144-149.

Özet – Günümüzde geliştirilen sistemler, karmaşık süreçler ve içerisinde birçok süreç değişkeni içermektedir. Bu durumda süreçlerin izlenmesi ve hata durumunda tespit edilmesi, süreçlerin aksamadan devam etmesi için zorunlu hale gelmiştir. Ancak süreç verilerini etkili şekilde izlemek, mevcut durumlarını analiz etmek, süreç anormalliklerini tespit ve teşhis etmek klasik yöntemlerle yapılamayacak kadar zor olduğundan daha etkili yöntemlere ihtiyaç duyulmaktadır.

Bu çalışmada, günlük yüksek miktarda veri üreten ve işleyen proseslere sahip tasarruf finansmanı sektöründe faaliyet gösteren firmalar baz alınarak, bir proses takip yazılımı önerisinde bulunulmuştur. Geliştirilen yazılımda, belirli periyotlarda ve belirlenen kural tabanlı analiz methodları kullanılarak proseslerin izlenmesi ve hata oluşması durumunda, anlık alarm üretilmesi gibi özellikler bulunmaktadır. Önerilen yazılım mimarisinin kullanılması durumunda servislerin düzenli çalışması ile sistem devamlılığın da sağlanacağı öngörülmektedir.

Anahtar Kelimeler – Proses Yönetimi, Proses Takibi, Kural Tabanlı Kontrol, Hata Yakalama, Yazılım Proses İyileştirme.

I. GİRİŞ

Yeni teknolojilerin gelişmesi sonucunda, eskiden daha çok kapalı olan sistem mimarilerini, günümüzde daha fazla dış sistem ile etkileşime zorlamıştır. Bunun bir sonucu olarak sistem mimari yapıları da büyümüştür. Giderek karmaşık bir yapı halini alan organizasyonlardaki sistemler beraberindeki uygulamalar ve cihazlar ile servis adı verilen yapılar üzerinden, çok sayıda günlük veri üretmeye başlamıştır. Dolayısıyla servis süreçlerin temel yapıtaşı haline gelmiş ve takibinin yapılması da elzem olmuştur. Servislerin takibi için büyük miktarda verinin analiz edilmesi ve geçerli bilgilerin çıkarılmasını gerektirmektedir. Üstelik, farklı iş süreçleri arasındaki etkileşim çok karmaşık olduğundan, birden fazla iş sürecini aynı anda analiz etme konusundaki kapsamlı yeteneklere sahip, tüm iş sürecinin kapsamlı bir görünümünü keşfedebilen çözümlere de ihtiyaç duyulmaktadır[1].

Büyüyen yapıların kontrol ve takibi insansı yöntemlerde mümkün olamayacak kadar zor hale geldiğinden, temel amacı, bu sistemlerin denetlenmesi sürecinin yürütülmesini içeren yeni yazılımlar üretilmeye başlanmıştır. Proses takip yazılımları olarak isimlendirilen bu yazılımlar, darboğazları, verimsizlikleri ve iyileştirilecek alanları belirlemeye çalışmaktadır. Gerçek zamanlı verilerin analiz edilmesi iş akışlarını kolaylaştırılmasında, kaynak tahsisini optimize etmekte ve katma değeri olmayan

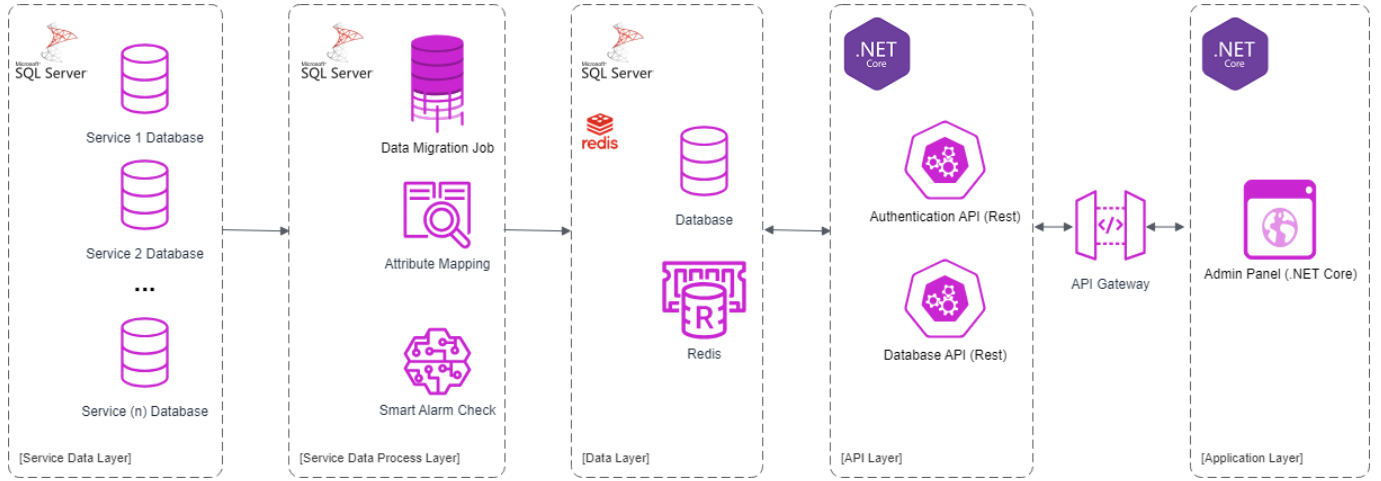
faaliyetleri ortadan kaldırmak amacıyla da kullanılabilir. Bu tür kontroller, süreç verimliliğini arttırmakta, döngü sürelerini azaltmakta ve üretkenliği artırarak kaynak kullanımını en üst düzeye çıkarmaktadır[2]. Ayrıca belirli göstergeleri sürekli izleyerek olası riskleri azaltmaktadır. Sorunların erken tespiti, proaktif müdahaleye olanak tanıyarak operasyonların sorunsuz ve kesintisiz olmasını da sağlamaktadır. Risk yönetimini, kurumsal esnekliği ve mevzuat uyumluluğunu artırır. Proseslere ait gerçek zamanlı bilgilerin sunulması entegre çalışan ve sistem üzerinde yapılan en küçük müdahalenin bile etkileyebileceği yapıların görülmesini ve sorunlara hızlı bir şekilde yanıt vermesine de olanak tanımaktadır. Bu durum, süreç devamlılığının sağlanması ile dinamik pazarlarda rekabet üstünlüğünü korumayı da desteklemektedir. Süreç izleme, potansiyel darboğazları ve gereksiz maliyetleri belirlenmesinde de etkilidir. Kuruluşlar, servislerin performans ölçümlerini analiz ederek, yeterince kullanılmayan kaynakları, yinelenen görevleri ve hataya açık süreçleri tespit edebilmektedirler. Bu bilgi, maliyet tasarrufu için süreç iyileştirmelerini, otomasyonu ve kaynak doğru tahsisinin sağlanmasında etkili olmaktadır. Optimize edilmiş süreçler israfı azaltılmasında ve maliyet verimliliğini artırarak kârlılığı arttırmaktadır. Süreç izleme yalnızca ölçülebilir faydaların yanı sıra, aynı zamanda sürekli iyileştirme ve yenilik kültürünü de teşvik etmektedir. Kazanılan bu içgörüler, kuruluşun bilgi tabanına katkıda bulunarak en iyi uygulamaları ve sürekli süreç iyileştirme için bir öğrenme kültürünü teşvik etmektedir[3].

Literatür çalışmaları incelendiğinde proses takip sistemleri üzerine yapılan araştırmaların yaygın olarak endüstride uygulama alanı bulduğu görülmektedir. Bunun nedeni çoğunlukla ilgili proseslerin ard arda işlemesi ve sonrasında gelecek proseslerin doğru çalışması için de önemli olması şeklinde ifade edilmektedir[3]. Çalışmaların proses takip kriterlerini oluşturması ve takibi işlemleri genellikle analitik methodlar veya yapay zeka modelleri ile sağlanmıştır[4], [5]. Yapılan bir çalışmada, belirsiz doğrusal zamanla değişmeyen sistemlerde hata tespitini gerçekleştirmek için sorunlu durumlar formülize edilerek doğrusal matris eşitsizlikleri oluşturulmuştur[6]. Diğer bir çalışmada izleme, hata teşhisi, işlem optimizasyonu ve kontrolü için geçmiş verilerden yararlanarak veriye dayalı modeller oluşturulmuştur. Uyarlanabilir aracı tabanlı çerçeve altyapısına sahip bu modeller çeşitli örnekler üzerinde simüle edilmiş ve sonuçları klasik modellerin sonuçlarıyla karşılaştırılmıştır[7]. Diğer bir çalışmada, ilaç endüstrisinde gerçek zamanlı olarak kullanılan proses verileri ile proses analitik teknolojisi adı verilen yöntem uygulanmıştır. Bu yöntem ile ilgili prostesteki sorunların anlaşılması, proses kalitesinin incelenmesi, anormal durumların tespiti, proses güvenilirliğinin iyileştirilmesi amaçlanmıştır[8]. Diğer bir çalışmada, kimyasal endüstrisindeki proseslerin izlenmesi için nedensel bağımlılık haritaları ve çok değişkenli istatistik methodlar tercih edilmiştir. Burada kullanılan veriseti Tennessee Eastman yöntemi adı verilen teknik ile elde edilmiş ve veriseti üzerinde temel bileşen analizi(PCA), kısmi küçük kareler yöntemi(PLS), değiştirilmiş mesafe (DI), değiştirilmiş nedensel bağımlılık (CD) gibi birçok yöntem uygulanmıştır[9]. Diğer bir çalışmada, reaktör proseslerinin takibi, sistem durumları ve durum uzayı parametreleri üzerinde kanonik değişken analizi(CVA) ve en küçük kareler yöntemi(PLS) kullanılarak gerçekleştirilmiştir[10]. Diğer bir çalışmada, makinelerin durumu ve arıza teşhisinin yapılabilmesine yönelik destek vektör makinesi(SVM) tabanlı bir yazılım önerilmiştir. Burada gerçekleştirilen proses takibi ile, makinelerin bakım maliyetlerinin azaltılması, üretkenliklerinin artması ve uzun süreli kullanımlarının sağlanması hedeflenmiştir[11]. Diğer bir çalışmada, imalat endüstrisinde üretim proseslerinin izlenebilmesi için operasyonel veriler üzerine kurulmuş olan yapay sinir ağı modelleri kullanılmıştır[12].

Bu çalışma kapsamında, günümüz yazılım entegratör araçlarından Rest ve SOAP mimarilerini kullanan servis yazılımları için, süreç izleme ve hata durumlarının tespitini sağlayan bir yazılım önerisinde bulunulmuştur. Önerilen yazılım, tasarruf finansmanı sektöründe faaliyet gösteren firmaların yüksek ölçekli veri trafiğini gerçekleştiren alt servis yazılımları üzerinde uygulanmıştır.

II. MATERYAL VE YÖNTEM

Geliştirilen servis prosesleri takip yazılımı için Şekil 1’de verilen sistem mimarisi oluşturulmuştur.



Şekil 1. Proses takip yazılımı sistem mimarisi

A. Veritabanı Sisteminin Oluşturulması

Proses takibinde, takibi yapılacak olan servislerin çalışma karakteristiğini gösteren uygun verilerin belirli periyotlarda çekilmesi gerekmektedir. Çalıştığımız uygulama servisleri incelendiğinde, yaptıkları işlemlerin kuyruk bilgilerini ve loglarını tuttukları tablolar MSSQL server veritabanı üzerinde bulunmaktadır. Buradaki bilgiler derlenerek servislerin çalışma, çalışmama veya hata durumları belirlenebilmektedir. Bunun için belirlenen bir excel formatında servislere dair bu durumları içeren listeler oluşturulmuştur. Liste örneği Tablo 1’de gösterilmektedir.

Tablo 1. Servis çalışmama ve hata kriterleri örnek listesi

Servis Adı	Çalışmama		Hata	
	Periyodu	Koşulu	Periyodu	
A	10 dk	10 dk dan daha fazla kuyrukta bekleyen kayıt olması	5 dk	Kuyrukta Status kodu 3 olan kayıt bulunması
B	5 dk	5 dk dan daha fazla kuyrukta bekleyen kayıt olması	3 dk	Kuyrukta ReturnResult E kodu olan kayıt bulunması
C	10 dk	Statu kodu 6 olan kayıtların kuyrukta 10 dk dan fazla beklemesi	5 dk	Kuyrukta Status kodu 3 olan kayıt bulunması

Hata oluşan veya çalışmayan bir serviste sorunun kaynağını bulma noktasında yapılacak analizlerde, ilk aşamada ilgili servisin veritabanına gidilerek sorgulama yapılması gerekmektedir. Kullanıcının ilk temel analizleri, geliştirilen uygulama içerisinde yapılabilmesi için, servislerin işlemlerine ait bazı bilgileri aktarmanın faydalı olacağı düşünülmüştür. Buna göre, belirlenen kriterler üzerinden servislerin yaptıkları işlemlere dair geçmiş log kayıtlarının alınarak, geliştirilen uygulama veritabanına özet şeklinde atılması tercih edilmiştir. Özet bilgilerin alınması ve işlenerek uygulama veritabanına aktarılması Tablo 1’de belirlenen periyotlardan daha küçük periyotlarda yapılması son derece önemlidir. Aksi durumda ilerleyen aşamalarda çalışacak olan uyarı mekanizması yeterli sürede alarm üretemeyecektir. Bunun için servis geçmişlerine dair özet bilgi alım periyotları Tablo 2’de görüldüğü şekilde belirlenmiştir.

Tablo 2. Servis özet bilgi alım periyodları

Servis Adı	Periyodu
A	5 dk
B	2 dk
C	5 dk

Servislerin geçmişleri her uygulama ve servisinde ayrı mantıkta oluşturulmuş tablolarda, hatta bazılarında dağıtık tablolarda tutulmaktadır. Bu nedenle farklı yapılarıdaki bu özet bilgileri uygulamamızın servis özetleri tablosunda ortaklaştırmamız gerekmektedir. Bu amaçla oluşturduğumuz mapping yapısı sayesinde farklı yapıları sahip herhangi bir servis sisteme eklenebilir hale getirilmiştir. Burada önemli görülen durumlardan birisi, özellikle büyük verilerin tutulduğu Json gibi alanların, tablo boyutunu arttırmaması için özet bilgiye dahil edilmemesidir. Oluşturulan mapping yapısına dair örnek Tablo 3’te gösterilmektedir.

Tablo 3. Servis tablosu ve özet bilgi tablosu eşleştirme örneği

Servis Adı	Alan Adı	Özet Alan Adı
A	RequestId	Ref_Id
A	Status	StatusId
A	JsonResult	-
A	TryCount	Ref_TryCount

Oluşturulan bir MSSQL job ile belirtilen yapı kullanılarak servislerin işlem bilgilerinin getirilmesi otomatik olarak sağlanmıştır. Job’ın çalışma zaman aralığı Tablo 2’de belirlenen referanslara bağlı kalınarak tüm süreleri kapsayabilmesi için en düşük referans süresinin yarısı olan 1dk olarak belirlenmiştir.

Sisteme ilk aşamada bağlanan 32 servisin ilerleyen aşamalarda 100’ün üzerine çıkacağı varsayılmaktadır. Sayısının artmasıyla birlikte servislerden elde edilen bilgilerin özet dahi olsa yüksek miktarları bulacağı öngörülmektedir. Bu durumda uygulama katmanının daha hızlı çalışabilmesi ve veritabanı düzeyinde yapacağı işlemleri en aza indirmek amacıyla bir caching mekanizması sisteme dahil edilmiştir. Kullanıcılar, kullanıcı rolleri, sayfa bilgileri, uygulama menüleri gibi sıklıkla değişmeyecek olan önyüze dair veriler caching yapısına dahil edilmiştir. Cache sistemi için günümüzde yaygın olarak kullanılan açık kaynak kodlu redis yapısı tercih edilmiştir[13].

B. Yazılım Sisteminin Oluşturulması

Servis takip uygulamasının birden fazla kullanıcısının olması ve internet üzerinden erişilebilmesi gibi ihtiyaçları bulunmaktadır. Web tabanlı olması istenen uygulama ve arkaplanda çalışacak servis yapıları için günümüzde yaygın olarak kullanılan teknolojilerden .Net Core Framework tercih edilmiştir. Yazılım geliştirme platformu olarak bu altyapıya uygun, Visual Studio 2021 ve programlama dili olarak da C# tercih edilmiştir.

Yazılım sistemi 2 ana bölümden oluşmaktadır. Birincisi kullanıcıların erişerek servis bilgilerini görebildiği ve analiz edebildiği önyüzü içeren admin panel uygulaması, ikincisi de arkaplan hizmetlerinin yapıldığı servis katmanıdır.

Admin panel bölümünde sistemin yönetilmesine ve takibine dair tüm fonksiyonlar ekranlar üzerinden kullanıma sunulmuştur. Admin panelde Tablo 4’te belirtilen fonksiyonlar bulunmaktadır.

Tablo 4. Admin panel fonksiyonları

Fonksiyon Adı	Açıklama
Login	Kullanıcı girişinin otomatik olarak yapıldığı ancak kullanıcı AD'de (Active Directory) tanımlı değilse manual giriş yapabildiği ekran
Yetkiler	Sayfa yetkilerinin tanımlandığı ekran
Roller	Yetkilerin gruplandığı ekran
Kullanıcılar	Sistemi kullanma hakkı bulunan kullanıcıların ve yetkilerinin tanımlandığı ekran
Servisler	Uygulamada tanımlı servislerin listelendiği ekran
Servis özet geçmişi	Servisler ekranında seçilen bir kayıda ait geçmiş servis işlemlerinin görüntülediği ekran
Sistem Parametreleri	Sistemin çalışması sırasında kullanılan parametrelerin ayarlanabildiği ekran
Servis alarm geçmişi	Servislerin alarm mekanizmasını tetikleyen geçmiş kayıtların görüntülediği ekran

Admin panelin kodlanması sürecinde tüm veritabanı işlemleri API adı verilen uygulama programlama arabirimleri üzerinden gerçekleştirilecek şekilde ayarlanmıştır. Bu durumda panel üzerinden doğrudan veritabanı erişimi söz konusu değildir. Bu şekilde API sistemlerinin dahip olduğu önemli bir güvenlik önlemi devreye alınmıştır hem de veritabanı işlemleri API tarafına yüklenerek kodlama düzeyi daha rahat hale getirilmiştir[14].

Yazılım işlemlerinin diğer bölümünü içeren Restful tabanlı servis katmanında 2 temel öge yer almaktadır. Restful mimarisinin seçilmesi, daha az veri transferinin yapıldığı için hızlı oluşu, birden fazla veri türünü ve güvenli kimlik doğrulamada yöntemlerini desteklemesi nedeniyle tercih edilmiştir[15]. Servis katmanındaki birinci bölümde kullanıcıların sisteme girişi sırasında login bilgilerinin sistemden alındığı ve otomatik login işlemlerinin yapıldığı bir erişim servisi bulunmaktadır. İkinci bölümde yazılım ön yüzünde bulunan ekranların ihtiyaç duyduğu veritabanı işlemlerini gerçekleştiren veritabanı servisi bulunmaktadır. Bu bölüm görevi itibariyle veritabanına doğrudan bağlanabilir ve uygulama katmanından gelen veritabanı taleplerine aracılık etmektedir.

III. SONUÇLAR

Bu çalışmada servislerin proses takibinin yapılabilmesi için bir yazılım önerisinde bulunulmuştur. Uygulama günümüzde kullanılan son teknoloji ürünleri temel alınarak oluşturulmuştur. Ayrıca uygulamanın geliştirilmesi sürecinde yapılması gereken veritabanı tasarımı ve yazılım bölümleri hakkında detaylı bilgi verilmiştir. Alarm ve uyarı yeteneklerinin de yazılıma eklenmesiyle yöneticileri önceden uyarı ve hızlı aksiyon alınmasına yardımcı olan özellikleri de bulunmaktadır.

Yapılan gerçek zamanlı sistem testleri sırasında; belirlenen servisleri yine belirlenen periyotlarda düzenli şekilde takip edebildiği, özet veri aktarımları sırasında ilgili sistemlerdeki kaynak kullanımını arttırmadığı görülmüştür. Ek olarak, uygulamanın standart sistem özelliklerine sahip sunucularda barındırılabilir kadar az kaynak kullanan bir uygulama olduğu da söylenebilir.

Önerilen sistem, servis tabanlı yazılımların olduğu her yerde kullanılması mümkün olan, esnek bir tasarım yapısına sahiptir. Bu mimari yeni özellikler eklenerek geliştirilmeye açık bir yapıdadır.

Çalışmanın ilerleyen aşamalarında yapay zeka modelleri de uygulamaya eklenerek sistem daha zeki hale getirilebilir. Özellikle alarm mekanizmasının, geçmiş örneklere bakılarak dinamik hale getirilmesi sağlanabilir, mevcut gidişattan yola çıkarak, servis işlem sürelerinin değişimi dikkate alınabilir ve takip süreçlerine eklenerek erken uyarı sistemi gibi yeni özellikler ile uygulama daha da gelişmiş hale getirilebilir.

KAYNAKLAR

- [1] C. Ji ve W. Sun, "A Review on Data-Driven Process Monitoring Methods: Characterization and Mining of Industrial Data", *Processes* 2022, Vol. 10, Page 335, c. 10, sy 2, s. 335, Şub. 2022, doi: 10.3390/PR10020335.
- [2] "The Power of Process Monitoring: Key Benefits & Steps". Erişim: 01 Temmuz 2024. [Çevrimiçi]. Erişim adresi: <https://www.boc-group.com/en/blog/bpm/the-power-of-process-monitoring/>
- [3] K. Severson, P. Chaiwatanodom, ve R. D. Braatz, "Perspectives on Process Monitoring of Industrial Systems", *IFAC-PapersOnLine*, c. 48, sy 21, ss. 931-939, Oca. 2015, doi: 10.1016/J.IFACOL.2015.09.646.
- [4] W. S. Chowdhury ve Y. Yan, "Applications of Artificial Intelligence to Instrumentation Systems for Monitoring Complex Industrial Processes", *Cybernetics and Intelligence*, Ara. 2023, doi: 10.26599/CAI.2024.9390006.
- [5] V. Uraikul, C. W. Chan, ve P. Tontiwachwuthikul, "Artificial intelligence for monitoring and supervisory control of process systems", *Eng Appl Artif Intell*, c. 20, sy 2, ss. 115-131, Mar. 2007, doi: 10.1016/J.ENGAPPAI.2006.07.002.
- [6] M. Zhong, S. X. Ding, J. Lam, ve H. Wang, "An LMI approach to design robust fault detection filter for uncertain LTI systems", *Automatica*, c. 39, sy 3, ss. 543-550, Mar. 2003, doi: 10.1016/S0005-1098(02)00269-8.
- [7] J. MacGregor ve A. Cinar, "Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods", *Comput Chem Eng*, c. 47, ss. 111-120, Ara. 2012, doi: 10.1016/j.compchemeng.2012.06.017.
- [8] T. Kourti, "The Process Analytical Technology initiative and multivariate process analysis, monitoring and control", *Anal Bioanal Chem*, c. 384, sy 5, ss. 1043-1048, Mar. 2006, doi: 10.1007/S00216-006-0303-Y.
- [9] L. H. Chiang ve R. D. Braatz, "Process monitoring using causal map and multivariate statistics: Fault detection and identification", *Chemometrics and Intelligent Laboratory Systems*, c. 65, sy 2, ss. 159-178, Şub. 2003, doi: 10.1016/S0169-7439(02)00140-5.
- [10] A. Simoglou, E. B. Martin, ve A. J. Morris, "Statistical performance monitoring of dynamic multivariate processes using state space modelling", *Comput Chem Eng*, c. 26, sy 6, ss. 909-920, Haz. 2002, doi: 10.1016/S0098-1354(02)00012-1.
- [11] A. Widodo ve B. S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis", *Mech Syst Signal Process*, c. 21, sy 6, ss. 2560-2574, Ağu. 2007, doi: 10.1016/j.ymsp.2006.12.007.
- [12] T. Herzog, M. Brandt, A. Trinchi, A. Sola, ve A. Molotnikov, "Process monitoring and machine learning for defect detection in laser-based metal additive manufacturing", *Journal of Intelligent Manufacturing* 2023 35:4, c. 35, sy 4, ss. 1407-1437, Nis. 2023, doi: 10.1007/S10845-023-02119-Y.
- [13] R. Chopade ve V. Pachghare, "A data recovery technique for Redis using internal dictionary structure", *Forensic Science International: Digital Investigation*, c. 38, s. 301218, Eyl. 2021, doi: 10.1016/J.FSIDI.2021.301218.
- [14] J. Ofoeda, R. Boateng, ve J. Effah, "Application programming interface (API) research: A review of the past to inform the future", *International Journal of Enterprise Information Systems*, c. 15, sy 3, ss. 76-95, Tem. 2019, doi: 10.4018/IJEIS.2019070105.
- [15] J. Meng, S. Mei, ve Z. Yan, "RESTful web services: A solution for distributed data integration", *Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, 2009, doi: 10.1109/CISE.2009.5365234.