

85/5/RPL-KCBR/PK.03.08/15/AGUSTUS/2024

**PERBANDINGAN PERFORMA *RUNTIME* JAVASCRIPT NODE.JS, BUN,
DAN DENO PADA REST API**

SKRIPSI

Diajukan untuk memenuhi sebagian syarat untuk memperoleh gelar Sarjana
Komputer pada Program Studi S1 Rekayasa Perangkat Lunak



Oleh

Septian Dwi Putra Pradipta

2007428

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK
KAMPUS UPI DI CIBIRU
UNIVERSITAS PENDIDIKAN INDONESIA
2024**

**PERBANDINGAN PERFORMA *RUNTIME* JAVASCRIPT NODE.JS, BUN,
DAN DENO PADA REST API**

Oleh

Septian Dwi Putra Pradipta

2007428

Sebuah skripsi yang diajukan untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer pada Program Studi Rekayasa Perangkat Lunak

© Septian Dwi Putra Pradipta 2024

Universitas Pendidikan Indonesia

Agustus 2024

Hak cipta dilindungi undang-undang.

Skripsi ini tidak boleh diperbanyak seluruhnya atau sebagian,
dengan dicetak ulang, difoto kopi, atau cara lainnya tanpa izin dari penulis.

ii

HALAMAN PENGESAHAN

Septian Dwi Putra Pradipta

**PERBANDINGAN PERFORMA *RUNTIME* JAVASCRIPT NODE.JS, BUN,
DAN DENO PADA REST API**

disetujui dan disahkan oleh pembimbing:

Pembimbing I



Dian Angraeni, S.ST., M.T.

NIP. 920190219930526201

Pembimbing II



Hendriyana, S.T., M.Kom.

NIP. 920230219960729201

Mengetahui,

Ketua Program Studi Rekayasa Perangkat Lunak



Mochamad Iqbal Ardimansyah, S.T., M.Kom.

NIP. 920190219910328101

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi dengan judul “PERBANDINGAN PERFORMA *RUNTIME* JAVASCRIPT NODE.JS, BUN, DAN DENO PADA REST API” ini beserta seluruh isinya adalah benar-benar karya saya sendiri. Saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika ilmu yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung risiko/sanksi apabila di kemudian hari ditemukan adanya pelanggaran etika keilmuan atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Bandung, 12 Agustus 2024



Septian Dwi Putra Pradipta

NIM 2007428

UCAPAN TERIMA KASIH

Puji syukur *alhamdulillah* kepada Allah SWT yang telah memberikan rahmat, karunia, dan hidayah-Nya, sehingga penulis mampu untuk menyelesaikan skripsi ini dengan judul “Perbandingan Performa *Runtime* Javascript Node.js, Bun, Dan Deno Pada Rest Api” untuk memperoleh gelar Sarjana Komputer pada Program Studi Rekayasa Perangkat Lunak di Universitas Pendidikan Indonesia.

Dalam proses penyusunan skripsi ini, penulis menyadari bahwa pencapaian ini sangat tidak lepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih untuk beberapa pihak berikut ini:

1. Ibu, Bapak, Adik dan keluarga besar penulis yang tidak pernah absen untuk mendukung, mendoakan, dan meridhoi selama masa perkuliahan, masa berlangsungnya penelitian ini, hingga penulis dapat menyelesaikan studi dan skripsi ini.
2. Bapak Mochamad Iqbal Ardimansyah, S.T., M.Kom., selaku Ketua Program Studi Rekayasa Perangkat Lunak dan selaku dosen wali.
3. Ibu Dian Anggraini, S.ST., M.T., selaku pembimbing pertama yang sangat berdedikasi dalam memberikan arahan dan bimbingan serta meluangkan waktu berharga selama penelitian.
4. Bapak Hendriyana, S.T., M.Kom. selaku pembimbing kedua yang juga senantiasa berdedikasi dalam memberikan arahan dan bimbingan serta meluangkan waktu berharga selama penelitian.
5. Seluruh dosen dan staff Program Studi Rekayasa Perangkat Lunak Universitas Pendidikan Indonesia Kampus Daerah Cibiru yang telah memberikan berbagai pengetahuan, keterampilan, dukungan dan arahan untuk terus berkembang serta bantuan akademik lainnya selama perkuliahan.
6. Seluruh rekan-rekan Rekayasa Perangkat Lunak angkatan 2020 dan rekan-rekan kontrakan yang telah kebersamai dalam proses perkuliahan bersama dengan penulis.

7. Terakhir, kepada diri saya sendiri yang telah berusaha sebaik mungkin dalam menempuh dan menyelesaikan pendidikan di Program Studi Rekayasa Perangkat Lunak Universitas Pendidikan Indonesia.

Penulis menyadari bahwasanya pada skripsi ini masih terdapat kekurangan. Dengan demikian, masukan dan kritik sangat diharapkan untuk perbaikan dan penyempurnaan skripsi ini. Penulis berharap, skripsi ini nantinya dapat menjadi acuan bagi penelitian di masa depan. Semoga usaha penulis mendapat berkah dari Allah SWT dan dapat memberikan manfaat bagi perkembangan ilmu serta berguna bagi masyarakat.

Bandung, 12 Agustus 2024

Septian Dwi Putra Pradipta

PERBANDINGAN PERFORMA *RUNTIME* JAVASCRIPT NODE.JS, BUN, DAN DENO PADA REST API

SEPTIAN DWI PUTRA PRADIPTA

2007428

ABSTRAK

REST API telah menjadi standar dalam pengembangan aplikasi web modern. JavaScript, bahasa pemrograman utama yang digunakan dalam pembuatan REST API, telah berkembang secara signifikan sejak pertama kali diperkenalkan pada tahun 1995. Node.js, salah satu *runtime* JavaScript yang paling populer, memungkinkan pengembang untuk mengelola aplikasi sisi server dan klien secara efisien dalam satu bahasa. Selain Node.js, dua *runtime* baru, yaitu Deno dan Bun, telah diperkenalkan sebagai alternatif yang menawarkan berbagai keunggulan, seperti peningkatan keamanan, transparansi, dan kinerja yang lebih baik. Penelitian ini bertujuan untuk menganalisis dan membandingkan performa REST API pada ketiga *runtime* tersebut, dengan fokus pada kecepatan respons, *throughput*, serta penggunaan sumber daya CPU dan memori. Penelitian ini didorong oleh klaim awal yang menjanjikan dari *runtime* Bun dan Deno, yang memerlukan evaluasi lebih mendalam dalam lingkungan uji yang terkontrol menggunakan *virtual machine* (VM). Metode statistik seperti ANOVA dan *Games-Howell* digunakan untuk menguji perbedaan performa secara signifikan. Hasil penelitian menunjukkan bahwa Deno memiliki performa terbaik dengan waktu respons dan Node.js *throughput* tertinggi, sementara Bun menunjukkan performa terendah dalam kedua aspek tersebut. Dalam hal penggunaan CPU, tidak ditemukan perbedaan signifikan antara ketiga *runtime*, namun Node.js menunjukkan efisiensi tertinggi dalam penggunaan memori. Temuan ini dapat menjadi pedoman penting bagi pengembang dalam memilih *runtime* yang paling sesuai untuk kebutuhan mereka.

Kata Kunci : REST API, Node.js, Deno, Bun, JavaScript, performa *runtime*

**PERBANDINGAN PERFORMA RUNTIME JAVASCRIPT NODE.JS, BUN,
DAN DENO PADA REST API**

SEPTIAN DWI PUTRA PRADIPTA

2007428

ABSTRACT

REST APIs have become a standard in modern web application development. JavaScript, the main programming language used in the creation of REST APIs, has evolved significantly since it was first introduced in 1995. Node.js, one of the most popular JavaScript runtimes, allows developers to efficiently manage server- and client-side applications in a single language. Besides Node.js, two new runtimes, Deno and Bun, have been introduced as alternatives that offer various advantages, such as increased security, transparency, and better performance. This research aims to analyze and compare the performance of REST APIs on these three runtimes, with a focus on response speed, throughput, and CPU and memory resource usage. This research is driven by the promising initial claims of the Bun and Deno runtimes, which require more in-depth evaluation in a controlled test environment using a virtual machine (VM). Statistical methods such as ANOVA and Games-Howell were used to test for significant performance differences. The results showed that Deno had the best performance with the highest response time and Node.js throughput, while Bun showed the lowest performance in both aspects. In terms of CPU usage, no significant difference was found between the three runtimes, but Node.js showed the highest efficiency in memory usage. These findings can be an important guideline for developers in choosing the most suitable runtime for their needs.

Keywords : REST API, Node.js, Deno, Bun, JavaScript, Performance runtime

DAFTAR ISI

HALAMAN PENGESAHAN	iii
LEMBAR PERNYATAAN	iv
UCAPAN TERIMA KASIH	v
ABSTRAK	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
1.5 Batasan Masalah	3
1.6 Struktur Organisasi Skripsi	3
BAB II KAJIAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 <i>Application Programming Interface</i>	9
2.2 <i>Representational State Transfer</i>	10
2.3 JavaScript	10
2.4 <i>Runtime JavaScript</i>	11
2.4.1 Node.js	11
2.4.2 Bun	12
2.4.3 Deno	13
2.5 ISO 25010	14
2.6 Uji Performa	15
2.6.1 K6	15
2.7 <i>Virtual Machine</i>	15
2.8 Docker	16

2.9 Server Monitoring	16
2.9.1 Grafana	16
2.9.2 Prometheus	17
2.9.3 Node Exporter	17
2.9.4 Influxdb	17
BAB III METODE PENELITIAN	19
3.1 Metode Penelitian	19
3.1.1 Klasifikasi Penelitian	20
3.1.2 Studi Deskriptif I	20
3.1.3 Studi Prespektif	20
3.1.4 Studi Deskriptif II	22
3.2 Alat dan Bahan Penelitian	22
3.3 Instrumen Penelitian	24
3.4 Prosedur Penelitian	25
3.5 Analisis Data	26
3.6 Hipotesis	26
BAB IV TEMUAN DAN PEMBAHASAN	28
4.1 Pengembangan REST pada <i>runtime</i> Node.js, Bun, dan Deno	28
4.2 Pra Pengujian	28
4.3 Persiapan Lingkungan Pengujian	29
4.3 Aspek <i>Time Behavior</i>	31
4.4 Aspek Penggunaan Sumber Daya	35
4.5 Pembahasan	40
BAB V SIMPULAN, IMPLIKASI, DAN REKOMENDASI	44
5.1 Simpulan	44
5.2 Implikasi	44
5.3 Rekomendasi	44
DAFTAR PUSTAKA	46
LAMPIRAN	51

DAFTAR TABEL

Tabel 2. 1.....	5
Tabel 3. 1 Spesifikasi API.....	20
Tabel 3. 2 Spesifikasi server runtime.....	23
Tabel 3. 3 Perangkat lunak yang digunakan.....	23
Tabel 4. 1 Kondisi awal server sebelum uji.....	29
Tabel 4. 2 Pengujian.....	29
Tabel 4. 3 Rekapitulasi hasil uji response time dan troughput.....	31
Tabel 4. 4 Rekapitulasi penggaan CPU dan memory.....	36
Tabel 4. 5 Rekapitulasi Uji Homogenitas.....	41
Tabel 4. 6 Rekapitulasi Uji Welch Anova.....	41
Tabel 4. 7 Rekap Hasil Games-Howell Test.....	42

DAFTAR GAMBAR

Gambar 2. 1 Event-driven architecture of Node.js (Feroj, 2023)	11
Gambar 2. 2 Node.js Ecosystem (Feroj, 2023)	12
Gambar 2. 3 Bun Ecosystem (Feroj, 2023).....	13
Gambar 2. 4 Deno Ecosystem	14
Gambar 2. 5 Kualitas Perangkat Lunak ISO 25010:2011	14
Gambar 3. 1 Metode Penelitian DRM.....	19
Gambar 3. 2 Arsitektur lingkungan pengujian server	21
Gambar 3. 3 Prosedur Penelitian.....	26
Gambar 4. 1 Grafana monitoring server runtime.....	28
Gambar 4. 2 Response time (biru) Node.js (hijau) Bun (merah) Deno	32
Gambar 4. 3 Komparasi response time	33
Gambar 4. 4 Throughput (biru) Node.js (hijau) Bun (merah) Deno.....	34
Gambar 4. 5 Komparasi throughput.....	35
Gambar 4. 6 Penggunaan CPU (biru) Node.js (hijau) Bun (merah) Deno.....	37
Gambar 4. 7 Komparasi penggunaan CPU	38
Gambar 4. 8 Penggunaan memory (biru) Node (hijau) Bun (merah) Deno.....	39
Gambar 4. 9 Komparasi penggunaan Memori	40

DAFTAR LAMPIRAN

Lampiran 1 Implementasi Sever runtime dan grafana pada docker.....	51
Lampiran 2 Hasil Uji K6.....	53
Lampiran 3 Bukti uji pengujian (video) dan metadata dari grafana	59

DAFTAR PUSTAKA

- Abbadini, M., Facchinetti, D., Oldani, G., Rossi, M., & Paraboschi, S. (2023a). Cage4Deno: A Fine-Grained Sandbox for Deno Subprocesses. *Proceedings of the ACM Asia Conference on Computer and Communications Security*, 149–162. <https://doi.org/10.1145/3579856.3595799>
- Abbadini, M., Facchinetti, D., Oldani, G., Rossi, M., & Paraboschi, S. (2023b). NatiSand: Native Code Sandboxing for JavaScript Runtimes. *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, 639–653. <https://doi.org/10.1145/3607199.3607233>
- AbhaAnand, B., & abhaanandb. (2021). *A survey on docker container and its use cases*. <https://api.semanticscholar.org/CorpusID:235742161>
- Aceituno-Rojo, M. R., Condori-Alejo, H. I., & Alzamora, G. S. (2021). Server monitoring model based on Balanced Scorecard and SNMP of an electric power company. *2021 IEEE XXVIII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 1–4. <https://doi.org/10.1109/INTERCON52678.2021.9532755>
- Afifah, F. N., Dewantara, N. P., Faiz, A. F., Romansyah, S., Negara, R. M., Mayasari, R., & Astuti, S. (2023). Cost-Effective Automation: Cloud-Based Monitoring Combining HPA with VPA for Scalable Startups. *2023 9th International Conference on Wireless and Telematics (ICWT)*, 1–5. <https://doi.org/10.1109/ICWT58823.2023.10335324>
- AkileshN, S., & Hallikar, R. S. (2022). *End to End Representational State Transfer Application Programming Interface Development*. <https://api.semanticscholar.org/CorpusID:251642346>
- Ardiansyah, H., & Fatwanto, A. (2022). Comparison of Memory usage between REST API in Javascript and Golang. *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 22(1), 229–240. <https://doi.org/10.30812/matrik.v22i1.1325>
- Baniaş, O., Florea, D., Gyalai, R., & Curiac, D.-I. (2021). Automated Specification-Based Testing of REST APIs. *Sensors*, 21(16), 5375–5375. <https://doi.org/10.3390/s21165375>
- Basumatary, B., & Agnihotri, N. (2022). Benefits and Challenges of Using NodeJS. *International Journal of Innovative Research in Computer Science & Technology*, 67–70. <https://doi.org/10.55524/ijircst.2022.10.3.13>
- Beermann, T., Alekseev, A., Baberis, D., Crépe-Renaudin, S., Elmsheuser, J., Glushkov, I., Svatos, M., Vartapetian, A., Vokac, P., & Wolters, H. (2020). Implementation of ATLAS Distributed Computing monitoring dashboards using InfluxDB and Grafana. *EPJ Web of Conferences*, 245, 03031. <https://doi.org/10.1051/epjconf/202024503031>

- Busro, S. A. B. C., Sucipto, Rina Firliana, M Najibuloh Muzzaki, Anita Sari Wardani, M Iqbal Khalid, Adhi Wicak Milbar Gamas, & Heru Setiawan. (2022). Rancangan Pembuatan API Website Data Tanaman Obat Dan Langka Kabupaten Kediri. *Bulletin of Information Technology (BIT)*, 3(4), 255–260. <https://doi.org/10.47065/bit.v3i4.373>
- Carbone, M., Lee, W., & Zamboni, D. (2020). Taming Virtualization. *IEEE Security & Privacy*, 6. <https://api.semanticscholar.org/CorpusID:14241494>
- De Macedo, J., Abreu, R., Pereira, R., & Saraiva, J. (2022). WebAssembly versus JavaScript: Energy and Runtime Performance. *2022 International Conference on ICT for Sustainability (ICT4S)*, 24–34. <https://doi.org/10.1109/ICT4S55073.2022.00014>
- Ehsan, A., Abuhaliqa, M. A. M. E., Catal, C., & Mishra, D. (2022a). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences*, 12(9), 4369. <https://doi.org/10.3390/app12094369>
- Ehsan, A., Abuhaliqa, M. A. M. E., Catal, C., & Mishra, D. (2022b). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences*, 12(9), 4369. <https://doi.org/10.3390/app12094369>
- Faasse, S., Bucek, J., & Schmidt, D. (2020). Out of Band Performance Monitoring of Server Workloads. *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, 4–11. <https://doi.org/10.1145/3358960.3375795>
- Farzat, F. de A., Barros, M. de O., & Travassos, G. H. (2021). Evolving JavaScript Code to Reduce Load Time. *IEEE Transactions on Software Engineering*, 47(8), 1544–1558. <https://doi.org/10.1109/TSE.2019.2928293>
- Feroj, M. F. (2023). *Javascript Runtime Performance Analysis: Node And Bun*.
- Huang, X. (2020). Research and Application of Node.js Core Technology. *2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, 1–4. <https://doi.org/10.1109/ICHCI51889.2020.00008>
- Jan, S.-L., & Shieh, G. (2020). On the Extended Welch Test for Assessing Equivalence of Standardized Means. *Statistics in Biopharmaceutical Research*, 12(3), 344–351. <https://doi.org/10.1080/19466315.2019.1654915>
- Jason A. Clark. (2020). Application Programming Interface. Dalam *The SAGE International Encyclopedia of Mass Media and Society*. SAGE Publications, Inc. <https://doi.org/10.4135/9781483375519.n38>
- Kemer, E., & Samli, R. (2019). Performance comparison of scalable rest application programming interfaces in different platforms. *Computer Standards & Interfaces*, 66, 103355. <https://doi.org/10.1016/j.csi.2019.05.001>

- Kniazev, I., & Fitiskin, A. (2023). Choosing The Right Javascript Runtime: An In-Depth Comparison Of Node. Js And Bun. *ECONOMIC SCIENCES*, 108, 72.
- Kumar, A., & Shrivastava, Dr. V. (2024). Modern Web Development Using Node.js and Comparative Analysis with PHP. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 08(03), 1–2. <https://doi.org/10.55041/IJSREM28676>
- Lawi, A., Benny L. E. Panggabean, & Takaichi Yoshida. (2021). Evaluating GraphQL and REST API Services Performance in a Massive and Intensive Accessible Information System. *Comput.*
- Lucas, C., & Woten, J. (2019). Virtual Machines with Dedicated Graphics for Software Development. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1568–1569. <https://doi.org/10.1109/CSCI49370.2019.00300>
- Marzullo, T., Dey, S., Long, N., Leiva Vilaplana, J., & Henze, G. (2022). A high-fidelity building performance simulation test bed for the development and evaluation of advanced controls. *Journal of Building Performance Simulation*, 15(3), 379–397. <https://doi.org/10.1080/19401493.2022.2058091>
- Mehdi, A., Bali, M. K., Abbas, S. I., & Singh, M. (2023). “Unleashing the Potential of Grafana: A Comprehensive Study on Real-Time Monitoring and Visualization.” *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–8. <https://doi.org/10.1109/ICCCNT56998.2023.10306699>
- Merelo-Guervós, J., García-Valdez, M., & Castillo, P. (2023). An Analysis of Energy Consumption of JavaScript Interpreters with Evolutionary Algorithm Workloads. *Proceedings of the 18th International Conference on Software Technologies*, 175–184. <https://doi.org/10.5220/0012128100003538>
- Mira Orisa, Ahmad Faisol, & Mochammad Ibrahim Ashari. (2023). Perancangan Website Company Profile Menggunakan Design Science Research Methodology (DSRM). *Jurnal Informatika Teknologi dan Sains*, 5(1), 160–164. <https://doi.org/10.51401/jinteks.v5i1.2576>
- Nurwasito, H., & Rahmawati, R. (2021). Weighted Response Time Algorithm for Web Server Load Balancing in Software Defined Network. *2021 3rd International Conference on Electronics Representation and Algorithm (ICERA)*, 143–148. <https://doi.org/10.1109/ICERA53111.2021.9538792>
- Pawar, S., K, S., & Laxmi, G. (2023). Application Programming Interface with a Case Study of SOA. *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, 1–5. <https://doi.org/10.1109/ICIICS59993.2023.10421593>

- Penkov, S., Taneva, A., & Kazala, R. (2020). Supervisory and Data Acquisition project with LoRa. *2020 International Conference Automatics and Informatics (ICAI)*, 1–4. <https://doi.org/10.1109/ICAI50593.2020.9311338>
- Prayogi, A. A., Niswar, M., Indrabayu, & Rijal, M. (2020). Design and Implementation of REST API for Academic Information System. *IOP Conference Series: Materials Science and Engineering*, 875(1), 012047. <https://doi.org/10.1088/1757-899X/875/1/012047>
- Riadi, S., Rukmayadi, D., Roswandi, I., & Wangitan, R. (2020). Pengaruh Perbedaan Dosis NaOH pada Pembuatan Sabun dengan Metode ANOVA Satu Arah dan Penentuan Perbandingan 3 Jenis Minyak sebagai Bahan Utama dengan Metode AHP pada Produk Sabun Mandi Ramah Lingkungan. *Jurnal Ilmiah Teknik Industri*, 8(2). <https://doi.org/10.24912/jitiuntar.v8i2.7356>
- Sargsyan, L., & Martins, F. (2020). Evaluation of a new visualization and analytics solution for slow control data for large scale experiments. *EPJ Web of Conferences*, 245, 07001. <https://doi.org/10.1051/epjconf/202024507001>
- Sharma, A., Patidar, S., Upadhyay, A., & Rajendra Acharya, U. (2019). Accurate tunable-Q wavelet transform based method for QRS complex detection. *Computers & Electrical Engineering*, 75, 101–111. <https://doi.org/10.1016/j.compeleceng.2019.01.025>
- Siddiqui, I., Pandey, A., Jain, S., Kothadia, H., Agrawal, R., & Chankhore, N. (2023). Comprehensive Monitoring and Observability with Jenkins and Grafana: A Review of Integration Strategies, Best Practices, and Emerging Trends. *2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 1–5. <https://doi.org/10.1109/ISMSIT58785.2023.10304904>
- Subramanian, H., & Raj, P. (2019). *Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs*. Packt Publishing Ltd.
- Sykora, M., Dolezal, Z., Kodys, P., & Kroll, J. (2022). Monitoring System of the ATLAS ITk Laboratory. *Journal of Physics: Conference Series*, 2374(1), 012098. <https://doi.org/10.1088/1742-6596/2374/1/012098>
- Vales, Z., & Brada, P. (2020). Service API Modeling and Comparison: A Technology-Independent Approach. *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 158–161. <https://doi.org/10.1109/SEAA51224.2020.00034>
- Wicha, M., & Pańczyk, B. (2023). Performance analysis of REST API technologies using Spring and Express.js examples. *Journal of Computer Sciences Institute*, 29, 352–359. <https://doi.org/10.35784/jcsi.3796>

- Wirfs-Brock, A., & Eich, B. (2020). JavaScript: the first 20 years. *Proceedings of the ACM on Programming Languages*, 4(HOPL), 1–1. <https://doi.org/10.1145/3386327>
- Yenugula, M., Kodam, R., & He, D. (2019). Performance and load testing: Tools and challenges. *International Journal of Engineering in Computer Science*, 1(1), 57–62. <https://doi.org/10.33545/26633582.2019.v1.i1a.102>
- Zadka, M. (2019). Docker. Dalam *DevOps in Python* (hlm. 147–150). Apress. https://doi.org/10.1007/978-1-4842-4433-3_12