

## **ABSTRACT**

*Containers are a virtualization technology development that is widely used to develop and deliver software for cloud computing services. Containers allow for efficient sharing of hosted resources between cloud tenant platforms, so containers are lighter and faster than virtual machines. In a container architecture, the container runtime is responsible for loading container images from the repository, monitoring local system resources, isolating system resources for container usage, and managing the container lifecycle. Hosts with a large number of containers cause a significant decrease in performance, both containers deployed on cloud tenant platforms and on local hosts. This issue occurs because the container runtime is responsible for running the containers on the host operating system. The choice of container runtime is very crucial to keep up with the increasing number of containers used. This research focuses on analyzing the performance of three container runtimes namely Containerd, CRI-O, and Kata Containers in Kubernetes orchestration. This research scenario uses scalability with a different number of containers, namely 10, 20, and 40 containers. The parameters to be analyzed are the performance of the CPU, memory, throughput, and latency. Based on the test results it is known that the container runtime Containerd has superior and optimal performance based on the results of throughput, latency, and memory parameters with the results of each parameter being 2301.75 MB/s, 0.359 ms, and 13001.8 MB/s.*

**Keywords:** *Container runtime, Containerd, CRI-O, Kata Containers, Kubernetes.*