

ABSTRACT

In In this ever-evolving digital age, operational efficiency and productivity are top priorities for businesses. One important aspect that is often overlooked is effective and integrated meeting room management. PT. Indo Trans Teknologi Bandung addresses this need by developing the INMEET application, a digital solution designed to optimize meeting room management by providing real-time information on room availability, booking schedules, and available facilities. The INMEET application integrates with various calendar systems, email, and other productivity applications, making it easier for users to book and manage meeting rooms efficiently.

The development of this application applies a Continuous Integration (CI) and Continuous Deployment (CD) approach supported by GitLab, Docker, and GitLab Runner. This system provides a comprehensive overview of the results of each automatic execution of the Continuous Integration/Continuous Delivery (CI/CD) process. Each row on this dashboard represents a single pipeline that has been run, which is a series of automated steps for building code, running tests, or preparing application releases as defined. Overall, the “Pipelines” dashboard is a vital tool for development teams to monitor project health, respond to issues quickly, and ensure continuous code quality and collaboration.

The “Status” column provides the most important indication of whether the pipeline was successful or failed. The “Passed” status, marked with a green checkmark icon, means that all stages in the pipeline have been successfully completed without errors. This indicates that the newly integrated code is stable, has passed all automated tests, and is ready for the next step in the deployment cycle, such as being tested in a simulated environment or even released to users. Conversely, the “Failed” status, marked with a red cross icon, indicates that at least one stage or job in the pipeline has failed. This serves as an early warning that there is an issue with the code, testing, or configuration, and requires investigation and resolution by the development team.

The system also provides other crucial information for each pipeline. The “Pipeline” column provides details such as a unique ID number (e.g., #15979), a description of the code change that triggered the pipeline (“Merge branch ‘NewFeature/CRD-10253/’”), and the commit ID (code version identifier). “Duration” displays the time required for the pipeline to complete, for example, 06:37 minutes for pipeline #15979, an important metric for measuring process efficiency. The “Created by” column shows who (the developer) initiated the pipeline execution, ensuring accountability within the team. Finally, the “Stages” column, which includes build, test, publish, release, and database, visually represents the stages executed in the pipeline with green icons for successful stages and red icons for failed stages, helping to identify where issues occurred if there are failures.