BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

Perancangan Infrastruktur Virtual Laboratory Multi-Layanan Berbasis Cloud Computing Menggunakan Metode NDLC

Metalika Gunena*, Anritsu Steven Christian Polii, Antonius P G Manginsela, Franky Manoppo, Olga Engelien Melo

Program Studi D-IV Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Manado, Manado, Indonesia
Email: [1,*]metalikagunena02@gmail.com, [2]anritsupolii@polimdo.ac.id, [3]anton@polimdo.ac.id, [4]franky.cliford@gmail.com, [5]olgameloaak@gmail.com
Email Penulis Korespondensi: metalikagunena02@gmail.com

Abstrak−Keterbatasan spesifikasi perangkat keras mahasiswa menjadi hambatan utama dalam pelaksanaan praktikum berbasis komputasi di lingkungan pendidikan tinggi, khususnya pada praktikum yang membutuhkan aplikasi dengan kebutuhan sumber daya komputasi tinggi, seperti perangkat lunak pemrograman, simulasi jaringan, hingga berbagai aplikasi penunjang praktikum lainnya, yang seringkali tidak dapat dijalankan secara optimal pada perangkat mahasiswa dengan spesifikasi rendah. Kondisi ini menyebabkan proses pembelajaran praktik menjadi tidak merata dan sangat bergantung pada kemampuan perangkat pribadi masing-masing mahasiswa. Berdasarkan permasalahan tersebut, Penelitian ini bertujuan merancang dan mensimulasikan sistem Virtual Laboratory berbasis cloud computing yang dapat diakses melalui web tanpa memerlukan spesifikasi perangkat keras tinggi di sisi pengguna, sekaligus memberikan kontribusi berupa arsitektur laboratorium virtual terintegrasi yang menggabungkan tiga model layanan cloud computing sekaligus, yaitu Infrastructure as a Service (IaaS), Platform as a Service (PaaS), dan Software as a Service (SaaS). Sebagai solusi, IaaS dibangun menggunakan Proxmox Virtual Environment (VE) sebagai hypervisor tipe-1, pfSense sebagai firewall virtual, Ubuntu Server sebagai host layanan, dan Docker sebagai platform kontainerisasi, dengan Cloudflare Tunnel berarsitektur Zero Trust untuk menjamin aksesibilitas dari luar jaringan kampus tanpa ketergantungan pada IP publik statis. Sistem dilengkapi autentikasi multi-peran (mahasiswa, dosen, admin) berbasis NIM/NIDN dan verifikasi OTP, modul Absensi berbasis kode QR yang terintegrasi dengan modul Timetable, serta portal utama berbentuk Progressive Web App (PWA) yang dapat dipasang pada perangkat pengguna. Pengembangan sistem menggunakan metode Network Development Life Cycle (NDLC). Hasil pengujian fungsional sementara menunjukkan seluruh layanan pada Kelas Pemrograman dan Kelas Jaringan, termasuk modul Absensi dan Timetable, berhasil diakses dari luar jaringan kampus, serta sistem autentikasi multi-peran berjalan sesuai rancangan.

Kata Kunci: Virtual Laboratory; IaaS; Proxmox VE; Cloudflare Tunnel; NDLC

Abstract−The limited hardware specifications of student devices remain a major obstacle in conducting computation-based practical courses in higher education, particularly in practicums that require applications with high computational resource demands, such as programming software, network simulation tools, and various other supporting applications, which often cannot run optimally on student devices with low specifications. This condition causes practical learning to become inconsistent and highly dependent on the capability of each student's personal device. Based on this problem, this research aims to design and simulate a cloud computing-based Virtual Laboratory system that can be accessed through a web browser without requiring high hardware specifications on the user side, while also contributing an integrated virtual laboratory architecture that combines three cloud computing service models simultaneously, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). As a solution, IaaS is built using Proxmox Virtual Environment (VE) as a type-1 hypervisor, pfSense as a virtual firewall, Ubuntu Server as the service host, and Docker as the containerization platform, along with Cloudflare Tunnel using a Zero Trust architecture to ensure accessibility from outside the campus network without dependence on a static public IP. The system is equipped with multi-role authentication (student, lecturer, admin) based on Student ID Number (NIM) or National Lecturer ID Number (NIDN) verification and OTP verification, a QR code-based Attendance module integrated with the Timetable module, and a main portal built as an installable Progressive Web App (PWA) on user devices. The system was developed using the Network Development Life Cycle (NDLC) method. Preliminary functional testing results show that all services in the Programming Class and Network Class, including the Attendance and Timetable modules, were successfully accessed from outside the campus network, and the multi-role authentication system functioned as designed.

Keywords: Virtual Laboratory; IaaS; Proxmox VE; Cloudflare Tunnel; NDLC

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat dalam era industri 4.0 mendorong kurikulum program studi Informatika untuk mengintegrasikan perangkat lunak yang bersifat resource-intensive, seperti simulasi jaringan, pengembangan aplikasi berbasis kontainer, dan manajemen basis data. Namun, tuntutan ini tidak sejalan dengan kondisi nyata di lapangan, di mana terdapat kesenjangan kemampuan ekonomi mahasiswa dalam pengadaan perangkat keras yang mumpuni. Mahasiswa dengan spesifikasi perangkat rendah kerap mengalami kendala teknis saat menjalankan instruksi praktikum yang berat. Di sisi lain, laboratorium komputer Program Studi Informatika masih beroperasi dalam paradigma sumber daya statis yang mengharuskan kehadiran fisik mahasiswa dan membatasi akses di luar jam operasional. Untuk mewujudkan sistem Virtual Laboratory, dibutuhkan dukungan infrastruktur server yang memadai; Unit Pelaksana Akreditasi Teknologi Informasi dan Komunikasi (UPA TIK) Politeknik Negeri Manado berperan sebagai penyedia layanan server kampus untuk keperluan ini.

Choudhary menunjukkan bahwa laboratorium praktikum berbasis kontainer Docker yang diorkestrasi dengan Kubernetes mampu memangkas waktu boot rata-rata dari 90 detik pada mesin virtual menjadi hanya 4,2 detik, mempercepat penyelesaian tugas praktikum sebesar 20,9%, serta meningkatkan tingkat kepuasan mahasiswa dari 3,2

Copyright © 2026 The Author, Page 1852
This Journal is licensed under a Creative Commons Attribution 4.0 International License

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

menjadi 4,3 pada skala 5. Temuan ini menunjukkan bahwa laboratorium berbasis kontainer dengan penskalaan dinamis secara signifikan meningkatkan kecepatan provisioning, efisiensi sumber daya, dan hasil pembelajaran, sekaligus mengurangi overhead infrastruktur [1]. Selain pada level kontainerisasi, penelitian terkait juga meninjau model layanan cloud computing pada laboratorium pendidikan dari berbagai sudut pandang. Zhang dan Guo melaporkan bahwa penerapan cloud computing pada laboratorium komputer kampus meningkatkan tingkat utilisasi ruang dari kisaran 50–68% (ruang tradisional) menjadi 79–92% (setelah menggunakan cloud computing), yang membuktikan bahwa cloud computing memiliki kesesuaian yang baik untuk diterapkan pada laboratorium komputer [2]. Zulfikar dkk. mengimplementasikan laboratorium virtual multi klaster berbasis Kubernetes untuk berbagi sumber daya antar-institusi, tetapi berfokus murni pada orkestrasi kontainer lintas klaster tanpa modul pendukung operasional laboratorium seperti presensi maupun penjadwalan [3]. Wahyuningsih dkk. mengevaluasi performa IaaS, PaaS, dan SaaS secara terpisah pada tiga penyedia cloud komersial berbeda dan menyimpulkan bahwa PaaS memberikan waktu respons terbaik sebesar 532,2 ms, namun ketiga model layanan tersebut diuji secara independen pada platform yang berbeda, bukan sebagai satu arsitektur yang terintegrasi [4].

Berdasarkan hasil kajian terhadap penelitian-penelitian terdahulu, dapat disimpulkan bahwa pengembangan Virtual Laboratory berbasis cloud computing masih didominasi oleh pendekatan yang berfokus pada satu aspek tertentu, seperti virtualisasi menggunakan Docker dan Kubernetes, evaluasi performa model layanan cloud secara terpisah, ataupun pengelolaan sumber daya komputasi. Meskipun masing-masing penelitian memberikan kontribusi penting, belum ditemukan penelitian yang mengintegrasikan ketiga model layanan cloud (IaaS, PaaS, dan SaaS) dalam satu arsitektur Virtual Laboratory yang juga mendukung kebutuhan operasional laboratorium pendidikan, seperti autentikasi multi-peran, presensi berbasis QR Code, penjadwalan praktikum, serta akses aman dari luar jaringan kampus menggunakan pendekatan Zero Trust melalui Cloudflare Tunnel. Kesenjangan tersebut menunjukkan bahwa masih diperlukan rancangan Virtual Laboratory yang tidak hanya menyediakan infrastruktur komputasi, tetapi juga mengintegrasikan layanan pembelajaran dan administrasi laboratorium dalam satu platform yang terpadu.

Sistem Virtual Laboratory berbasis Infrastructure as a Service (IaaS) yang memungkinkan orkestrasi sumber daya server terpusat. Berbeda dari pendekatan konvensional yang menyediakan mesin virtual per mahasiswa, penelitian ini mengadopsi arsitektur berbasis kontainer menggunakan Docker, sehingga layanan praktikum dapat diakses langsung melalui peramban web tanpa instalasi client tambahan; port kontainer dipetakan untuk menghindari benturan port, dan platform lab dikelola melalui manajemen penempatan kontainer serta antarmuka web yang dapat diakses pengguna. Untuk menjamin aksesibilitas dari luar jaringan kampus tanpa ketergantungan pada IP publik statis, penelitian ini mengimplementasikan Cloudflare Tunnel dengan arsitektur Zero Trust yang membuat koneksi terenkripsi dari server lokal ke jaringan DNS Cloudflare [5]. dilengkapi sistem autentikasi mandiri berbasis Nomor Induk Mahasiswa (NIM) dengan verifikasi OTP. Pengembangan sistem menggunakan metode Network Development Life Cycle (NDLC) yang terstruktur.

Penelitian ini bertujuan untuk: (1) merancang arsitektur Virtual Laboratory yang mengintegrasikan IaaS, PaaS, dan SaaS dalam satu portal akses tunggal yang dapat diakses melalui web dari mana saja; (2) mengimplementasikan layanan praktikum berbasis Docker untuk dua kelas pada tahap simulasi, yaitu Kelas Pemrograman dan Kelas Jaringan; (3) mengintegrasikan modul Absensi berbasis kode QR dan modul Timetable ke dalam portal sebagai pendukung operasional laboratorium; dan (4) membangun sistem autentikasi multi-peran berbasis NIM/NIDN dengan verifikasi OTP untuk menjamin keamanan akses. Penambahan kelas lainnya direncanakan setelah sistem bermigrasi ke server UPA TIK.

Infrastructure as a Service (IaaS) memberikan konsumen kapabilitas untuk melakukan provisioning terhadap processing, storage, network, dan sumber daya komputasi fundamental lainnya tanpa mengelola infrastruktur cloud yang mendasarinya. Platform as a Service (PaaS) memberikan kapabilitas untuk men-deploy aplikasi menggunakan bahasa pemrograman, pustaka, dan tools yang didukung penyedia, dengan kontrol terbatas pada aplikasi yang di-deploy. Software as a Service (SaaS) memberikan kapabilitas untuk menggunakan aplikasi siap pakai penyedia yang diakses melalui thin client seperti peramban web, tanpa kontrol atas infrastruktur maupun aplikasi yang mendasarinya [6][7]. Ketiga model layanan tersebut dapat hadir secara bersamaan dalam satu sistem dengan konsumen berbeda pada tiap lapisan: pada penelitian ini, lapisan IaaS dikonsumsi oleh peneliti selaku administrator yang melakukan provisioning mesin virtual di atas hypervisor Proxmox VE; lapisan PaaS dikonsumsi oleh mahasiswa melalui PHP/HTML Runner dan GNS3 untuk men-deploy karya sendiri tanpa mengelola server di baliknya; sedangkan lapisan SaaS dikonsumsi oleh mahasiswa dan dosen melalui layanan siap pakai seperti VS Code, phpMyAdmin, Absensi, dan Timetable yang diakses sepenuhnya melalui peramban web.

Penelitian ini menawarkan kontribusi berupa arsitektur Virtual Laboratory terpadu yang menggabungkan IaaS, PaaS, dan SaaS dalam satu portal berbasis web, dilengkapi autentikasi multi-peran, modul presensi berbasis QR Code, modul Timetable, serta mekanisme akses eksternal menggunakan Cloudflare Tunnel. Integrasi berbagai komponen tersebut diharapkan dapat mendukung penyelenggaraan praktikum yang lebih fleksibel, aman, dan mudah dikelola di lingkungan Politeknik Negeri Manado.

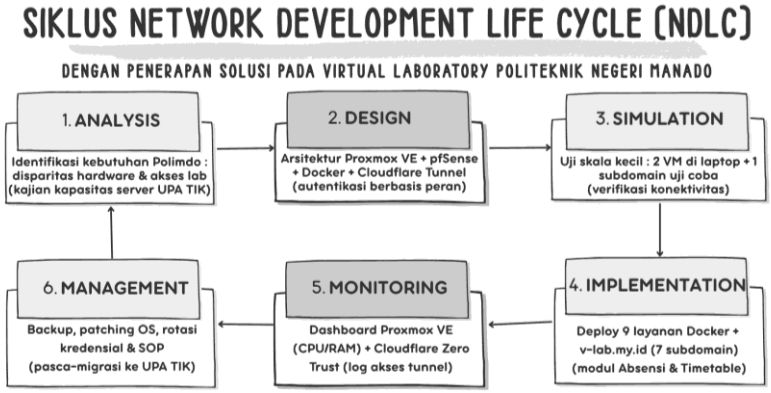
2. METODOLOGI PENELITIAN

Salah satu pendekatan metodologis yang dapat digunakan untuk memastikan keberhasilan pengembangan jaringan adalah Network Development Life Cycle (NDLC). Metode NDLC menawarkan kerangka kerja yang sistematis dan terstruktur

Copyright © 2026 The Author, Page 1853
This Journal is licensed under a Creative Commons Attribution 4.0 International License

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

melalui enam tahapan utama: inisiasi, analisis, desain, implementasi, operasi, dan pemeliharaan. Setiap tahapan dirancang untuk membimbing proses pengembangan jaringan dari awal hingga tahap operasional secara berkelanjutan [8]. Metode NDLC yang digunakan dalam penelitian ini memiliki 6 tahapan, yaitu Analysis (Analisis), Design (Desain), Simulation (Simulasi), Implementation (Implementasi), Monitoring (Pemantauan), dan Management (Manajemen) [9]. Keenam tahapan tersebut digambarkan secara siklus pada Gambar 1, sesuai karakteristik NDLC yang dirancang untuk mendukung evaluasi dan penyesuaian berulang selama pengembangan jaringan berlangsung.



Gambar 1. Diagram Alur Metode Network Development Life Cycle (NDLC)

Sebagaimana ditunjukkan pada Gambar 1, keenam tahapan tersebut digambarkan sebagai siklus berkelanjutan yang dimulai dari Analysis, berlanjut ke Design, Simulation, Implementation, dan Monitoring, hingga Management, sebelum kembali ke tahap Analysis untuk iterasi peningkatan berikutnya. Alur siklis ini menegaskan bahwa penerapan NDLC pada penelitian ini tidak berhenti pada satu putaran implementasi, melainkan dapat terus disempurnakan seiring kebutuhan operasional Virtual Laboratory berkembang, misalnya penambahan kelas praktikum baru atau migrasi ke infrastruktur produksi.

2.1 Tahapan Penelitian *(Analysis)*

Tahap analisis dilakukan melalui studi lapangan di Politeknik Negeri Manado. Hasil analisis mengidentifikasi dua permasalahan utama: (1) disparitas spesifikasi perangkat keras mahasiswa yang menghambat pelaksanaan praktikum resource-intensive seperti simulasi jaringan (GNS3) dan pengembangan berbasis container; (2) keterbatasan akses laboratorium yang terikat pada kehadiran fisik dan jam operasional, tanpa solusi akses jarak jauh yang memadai. Analisis juga mencakup kajian kapasitas infrastruktur server UPA TIK sebagai calon penyedia sumber daya komputasi, serta penentuan spesifikasi perangkat keras minimum untuk menjalankan simulasi awal, sebagaimana dirangkum di Tabel 1.

Tabel 1. Spesifikasi Perangkat Keras Server Simulasi

Komponen	Spesifikasi
CPU	2x AMD E2-9000e RADEON R2, 4 Cores
RAM	7,22 GB
Storage	87,92 GB
Hypervisor	Proxmox VE 9.1.1
Kernel	Linux 6.17.2

Pada Tabel 1, spesifikasi perangkat keras yang digunakan pada tahap simulasi meliputi CPU 2x AMD E2-9000e RADEON R2 dengan 4 core, RAM 7,22 GB, dan storage 87,92 GB, dengan Proxmox VE 9.1.1 berbasis kernel Linux 6.17.2 sebagai hypervisor. Spesifikasi ini merepresentasikan skala uji coba pada perangkat keras terbatas, bukan server kelas produksi, sehingga hasil pengujian pada tahap simulasi berfungsi sebagai validasi awal sebelum migrasi ke infrastruktur produksi UPA TIK dengan spesifikasi yang lebih memadai.

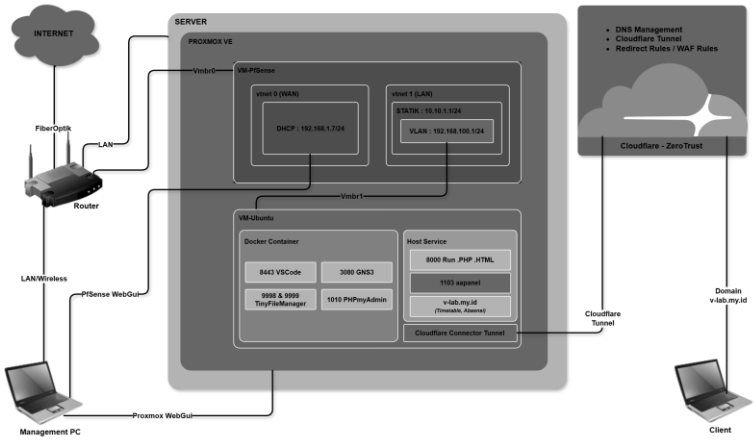
2.2 Perancangan Arsitektur *(Design)*

Arsitektur sistem dirancang dengan pendekatan berlapis menggunakan tiga komponen utama: (1) Proxmox VE sebagai hypervisor tipe-1 untuk orkestrasi sumber daya server terpusat; (2) pfSense sebagai gateway firewall virtual yang memisahkan jaringan WAN dan LAN internal; (3) Ubuntu Server sebagai host seluruh layanan kontainer Docker. Kemampuan Proxmox VE untuk mengelola berbagai jenis server virtual dengan sistem manajemen yang terpusat dan fleksibel. Teknik virtualisasi server berbasis Proxmox VE dapat mengurangi penggunaan hardware, memaksimalkan kapasitas CPU dan memori, serta menyederhanakan manajemen sumber daya [10]. pfSense Community Edition (CE) adalah platform routing, firewall, dan layanan jaringan open-source yang banyak digunakan oleh perusahaan dan organisasi dengan berbagai ukuran, tingkat kompleksitas, dan kebutuhan [11]. Docker adalah salah satu teknologi virtualisasi berbasis container yang paling dominan dan banyak digunakan, dan karena merupakan proyek open source,

Copyright © 2026 The Author, Page 1854
This Journal is licensed under a Creative Commons Attribution 4.0 International License

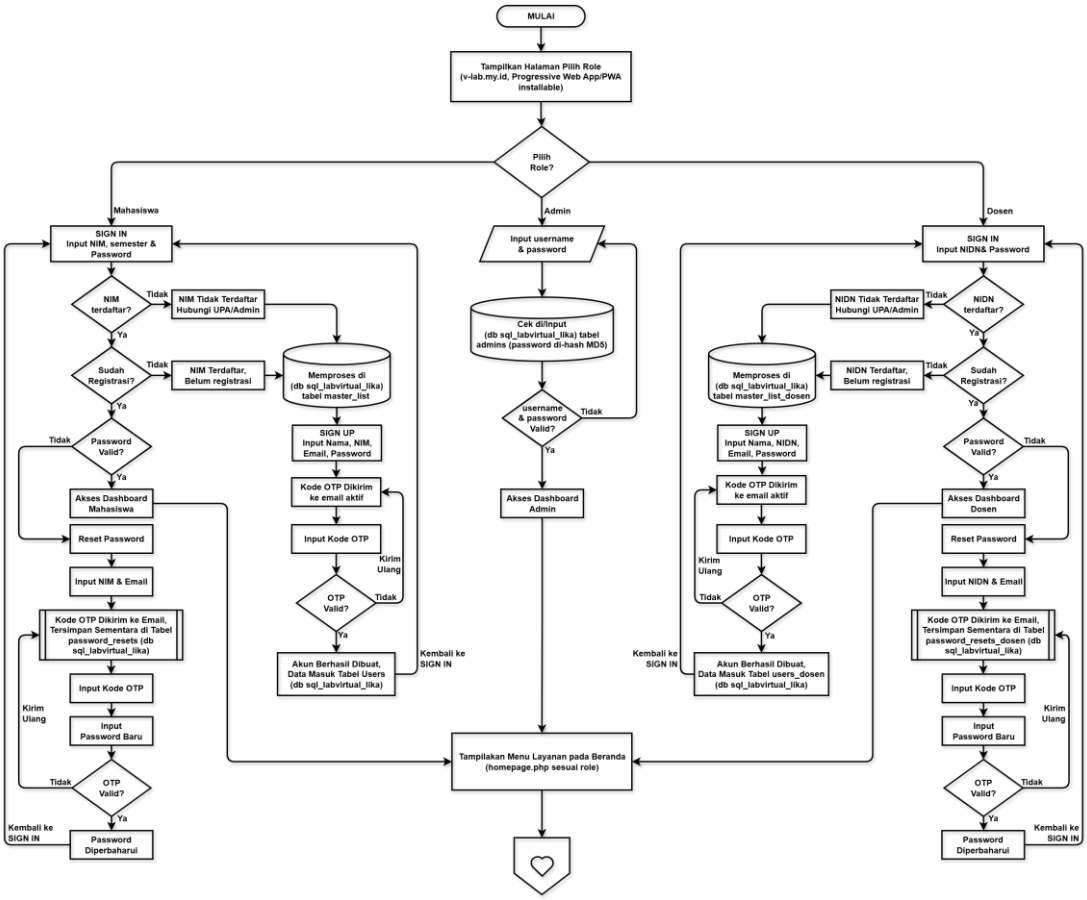
BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

Docker dapat digunakan untuk mengembangkan, menjalankan, dan mendeploy aplikasi secara efisien [12]. Rancangan arsitektur berlapis ini divisualisasikan pada Gambar 2.



Gambar 2. Topologi Arsitektur Sistem Virtual Laboratory

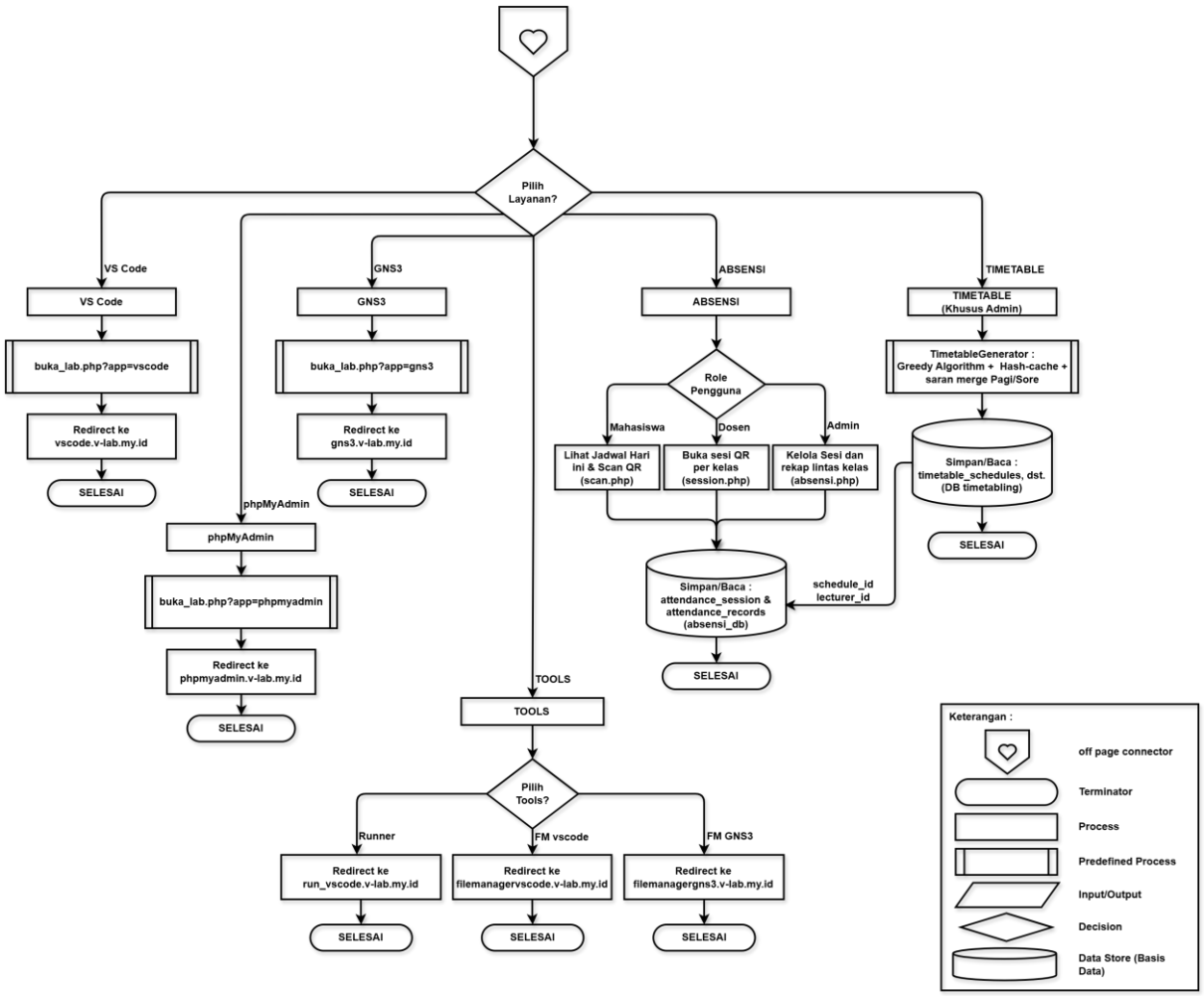
Gambar 2 memperlihatkan arsitektur jaringan berlapis, di mana pfSense berfungsi sebagai gateway yang menyaring seluruh lalu lintas pengguna sebelum mencapai Ubuntu Server dan layanan kontainer Docker di dalamnya, sehingga domain jaringan dan domain aplikasi terpisah secara tegas sesuai prinsip defense in depth. Pemisahan ini diwujudkan melalui dua jalur jaringan virtual dengan peran berbeda: satu jalur menghubungkan pfSense ke jaringan eksternal (WAN) dengan alamat yang diperoleh secara dinamis, sedangkan jalur lainnya menjadi penghubung internal antara pfSense dan Ubuntu Server dengan alamat tetap serta segmentasi jaringan tersendiri, sehingga lalu lintas internal tidak bercampur dengan lalu lintas eksternal. Agar dapat diakses dari luar jaringan kampus tanpa bergantung pada IP publik statis, sistem ini memanfaatkan Cloudflare Tunnel dengan pendekatan Zero Trust, di mana daemon cloudflared membangun koneksi outbound sehingga IP asli server tetap tersembunyi dan risiko serangan berbasis IP dapat diminimalkan. Sebagai bagian dari tahap perancangan (design) pada metode NDLC, Gambar 3 & 4 menunjukkan rancangan alur autentikasi dan pemilihan layanan pada sistem V-Lab.



Gambar 3. Flowchart Alur Autentikasi dan Pemilihan Layanan Sistem V-Lab

Copyright © 2026 The Author, Page 1855
This Journal is licensed under a Creative Commons Attribution 4.0 International License

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

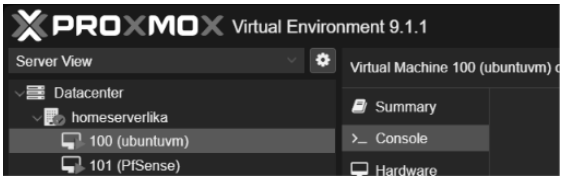


Gambar 4. Flowchart Alur Autentikasi dan Pemilihan Layanan Sistem V-Lab

Rancangan pada Gambar 3 & 4 mencakup dua bagian utama: alur autentikasi berdasarkan role pengguna (mahasiswa, dosen, admin) dengan validasi NIM/NIDN/password, sign up melalui verifikasi OTP, dan reset password bagi pengguna terdaftar; serta alur pemilihan layanan setelah login, di mana pengguna diarahkan ke masing-masing layanan (VS Code, GNS3, phpMyAdmin, Absensi, Timetable) sesuai hak akses role-nya. Rancangan ini menjadi dasar logika sistem yang selanjutnya diimplementasikan pada tahap berikutnya.

2.3 Simulasi *(Simulation)*

Sebelum instalasi massal, rancangan diuji terlebih dahulu dalam skala kecil pada perangkat keras laptop sebagai server simulasi. Pada tahap ini dilakukan instalasi Proxmox VE, kemudian dua mesin virtual dibuat: VM pfSense sebagai gateway dan VM Ubuntu Server sebagai host layanan, dengan Docker diinstal sebagai platform kontainerisasi. Kedua mesin virtual tersebut dapat dilihat pada Datacenter Proxmox VE, sebagaimana ditunjukkan pada Gambar 5.



Gambar 5. Tampilan Datacenter Proxmox VE dengan VM ubuntuvm dan VM PfSense

Gambar 5 menunjukkan node Proxmox VE (homeserverlika) pada Datacenter yang berisi dua mesin virtual sesuai rancangan, yaitu VM 100 (ubuntuvm) sebagai host layanan Docker dan VM 101 (PfSense) sebagai gateway firewall virtual, yang mengonfirmasi bahwa kedua komponen telah berhasil dibuat sebelum tahap pengujian konektivitas dilakukan. Di dalam VM 100 (ubuntuvm) tersedia seluruh layanan, baik yang dijalankan melalui Docker maupun yang diinstal secara manual sebagai host service, serta terpasang konektor Cloudflare Zero Trust yang memungkinkan seluruh layanan lokal tersebut diakses secara online. Cloudflare Zerotrust berperan penting dalam mengatasi keterbatasan IP publik, memungkinkan server diakses melalui domain pribadi meskipun layanan seperti FTP dan SSH hanya dapat dioperasikan di jaringan lokal [5].

Copyright © 2026 The Author, Page 1856
This Journal is licensed under a Creative Commons Attribution 4.0 International License

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

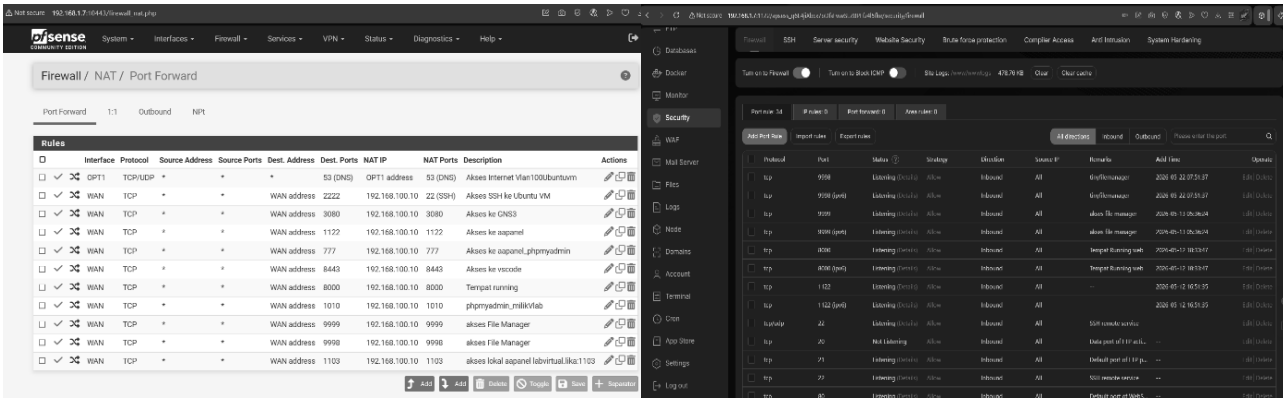
2.4 Implementasi Layanan (Implementation)

Tahap implementasi merupakan eksekusi langsung dari rancangan layanan yang telah melalui simulasi menggunakan Docker Compose untuk dua kelas laboratorium, sebagaimana dirangkum pada Tabel 2.

Tabel 2. Layanan Laboratorium Virtual yang Diimplementasikan

Layanan	Fungsi	Port	Kelas
VS Code	Editor kode berbasis web	8443	Pemrograman
phpMyAdmin	Manajemen basis data	1010	Pemrograman
PHP Web Server	Menjalankan file PHP/HTML	8000	Pemrograman
TinyFileManager	File manager berbasis web	9999	Pemrograman
GNS3	Simulasi jaringan	3080	Jaringan
TinyFileManager	File manager berbasis web	9998	Jaringan
aaPanel	Panel kontrol web server	1103	Manajemen
Absensi	Presensi berbasis kode QR per sesi kelas	(terintegrasi portal utama)	Lintas Kelas
Timetable	Manajemen jadwal praktikum (algoritma greedy + penggabungan Pagi/Sore)	(terintegrasi portal utama)	Lintas Kelas

Sebagaimana dirangkum pada Tabel 2, sembilan layanan diimplementasikan dan dikelompokkan ke dalam tiga kelas praktikum: Pemrograman, Jaringan, dan Manajemen, dengan Absensi dan Timetable sebagai layanan lintas kelas yang diakses langsung melalui portal utama tanpa memerlukan port maupun subdomain khusus. Konfigurasi pfSense pada penelitian ini dibatasi hanya fitur minimum yang diperlukan untuk packet forwarding antara interface WAN dan LAN yang dipertahankan, dengan aturan firewall dan konfigurasi NAT yang identik di semua sistem [11]. Pengaturan ini diterapkan pada dua lapisan sesuai alur trafik, yaitu NAT Port Forward pada pfSense sebagai gateway terluar dan firewall aaPanel di sisi Ubuntu Server, sebagaimana ditunjukkan pada Gambar 6.



Gambar 6. Konfigurasi NAT Port Forward pfSense dan Firewall aaPanel pada Ubuntu Server: (a) Port Forward pfSense; (b) Port Rule aaPanel

Gambar 6(a) menunjukkan konfigurasi Port Forward pada pfSense (Firewall > NAT > Port Forward) yang memetakan setiap port WAN ke IP internal 192.168.100.10, mencakup SSH (2222→22), GNS3 (3080), aaPanel (1122), phpMyAdmin aaPanel (777), VS Code (8443), file manager (9999 dan 9998), layanan running (8000), phpMyAdmin VLab (1010), dan akses lokal aaPanel labvirtual.lika (1103). Setelah diteruskan pfSense, trafik disaring kembali oleh firewall aaPanel pada Ubuntu Server, seperti ditunjukkan pada Gambar 6(b), yang mengizinkan (Allow) akses masuk (inbound) hanya pada port yang sesuai dengan layanan yang berjalan. Kedua lapisan ini bekerja berurutan mengikuti alur trafik dari WAN menuju layanan, sehingga permintaan akses harus lolos dari NAT pfSense terlebih dahulu sebelum difilter kembali oleh firewall aaPanel di level host, membentuk mekanisme keamanan berlapis (defense in depth) yang lebih kuat dibandingkan hanya mengandalkan satu lapisan firewall.

Untuk aksesibilitas dari luar jaringan kampus, Cloudflare Tunnel diimplementasikan dengan memanfaatkan satu domain utama v-lab.my.id, dengan daemon cloudflared pada Ubuntu Server membuka koneksi keluar terenkripsi ke edge server Cloudflare tanpa memerlukan pembukaan port pada firewall, sebagaimana dipetakan pada Tabel 3. Pendekatan ini sejalan dengan prinsip Zero Trust Network Access (ZTNA) “never trust, always verify”, yang mensyaratkan verifikasi identitas ketat dan hak akses minimal pada setiap permintaan layanan [13].

Tabel 3. Pemetaan Subdomain Layanan via Cloudflare Tunnel

Subdomain	Layanan
v-lab.my.id	Portal Utama (Login)
vscode.v-lab.my.id	VS Code
phpmyadmin.v-lab.my.id	phpMyAdmin

Copyright © 2026 The Author, Page 1857
This Journal is licensed under a Creative Commons Attribution 4.0 International License

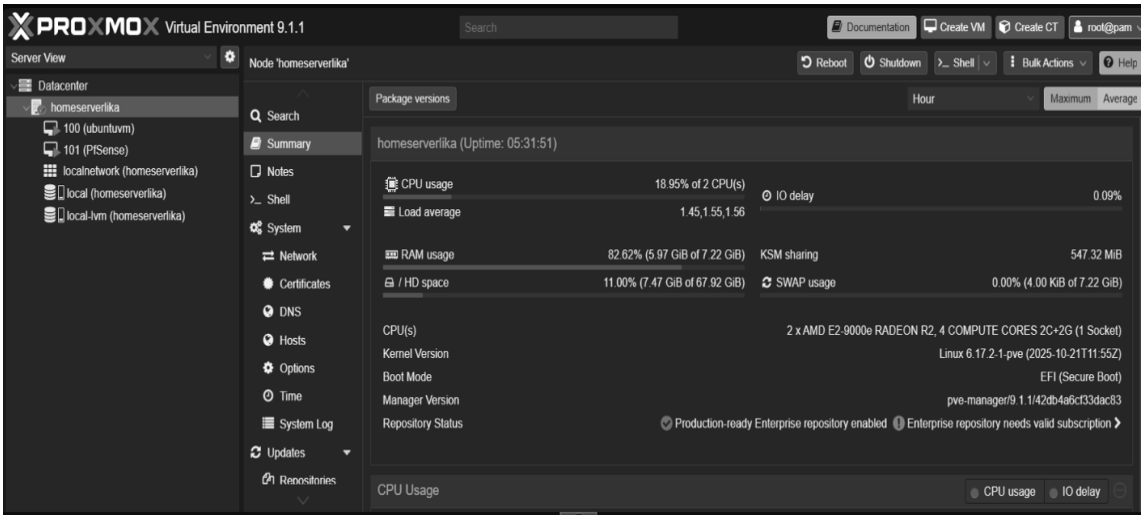
BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

Subdomain	Layanan
gns3.v-lab.my.id	GNS3
run_vscode.v-lab.my.id	PHP/HTML Runner
filemanagervscode.v-lab.my.id	File Manager VSCode
filemanagergns3.v-lab.my.id	File Manager GNS3

Sebagaimana dipetakan pada Tabel 3, tujuh subdomain dikonfigurasi melalui Cloudflare Tunnel: satu subdomain utama sebagai portal login dan enam subdomain layanan yang masing-masing diarahkan ke port kontainer Docker yang sesuai. Berbeda dengan layanan container, modul Absensi dan Timetable tidak memerlukan subdomain karena diakses langsung melalui portal utama dan divalidasi melalui sesi PHP yang sama.

2.5 Monitoring

Setelah jaringan aktif, sistem dipantau secara terus-menerus untuk memastikan kinerja berjalan optimal, mencakup pemantauan beban kerja CPU, penggunaan memori, serta kestabilan lalu lintas data. Pada penelitian ini, monitoring dilakukan melalui dashboard bawaan Proxmox VE yang menampilkan utilisasi CPU dan RAM secara real-time untuk setiap VM, sebagaimana ditunjukkan pada Gambar 7. Semakin besar jumlah pengguna, semakin besar pula peningkatan CPU dan throughput hingga titik tertentu, yang diikuti dengan penurunan tingkat availability [14].



Gambar 7. Tampilan Dashboard Proxmox VE untuk Monitoring Infrastruktur

Berdasarkan Gambar 7, dashboard Proxmox VE menunjukkan beban CPU node homeserverlika sebesar 49,83% dan penggunaan RAM 24,01% (1,73 dari 7,22 GiB), dengan SWAP 0% dan IO delay 0,17%, yang mengindikasikan resource host masih optimal dan mencukupi untuk menjalankan VM Ubuntu Server dan pfSense secara bersamaan.

2.6 Manajemen (Management)

Tahap management adalah perawatan harian agar sistem tetap berjalan lancar, mencakup: (1) backup data mahasiswa dan konfigurasi sistem secara berkala; (2) pembaruan keamanan (patching OS) pada Ubuntu Server dan Proxmox VE; (3) pengelolaan basis data NIM mahasiswa saat ada penambahan atau kelulusan mahasiswa; (4) dokumentasi setiap perubahan konfigurasi jaringan dan Docker Compose. Pada tahap simulasi, manajemen dilakukan secara manual; setelah migrasi ke server UPA TIK, diperlukan prosedur manajemen yang lebih formal dan terdokumentasi, mengingat kesalahan dalam manajemen jaringan dapat berdampak signifikan terhadap keberlangsungan operasional dan layanan yang diberikan [15].

Kebutuhan manajemen jangka panjang lain yang teridentifikasi mencakup rotasi kredensial dan token akses secara berkala (khususnya token Cloudflare Tunnel dan kata sandi akun admin), serta penyusunan dokumentasi prosedur operasional standar (SOP) untuk kebutuhan seperti penambahan kelas praktikum baru dan reset akun mahasiswa; kedua kebutuhan ini direncanakan untuk ditindaklanjuti pada tahap migrasi ke server produksi UPA TIK.

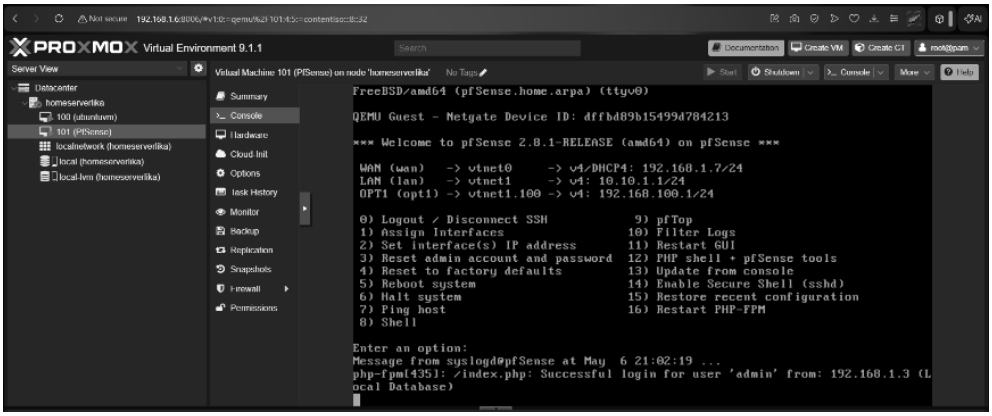
3. HASIL DAN PEMBAHASAN

3.1 Infrastruktur Sistem

Instalasi Proxmox VE berhasil dilakukan pada perangkat keras yang tersedia. Dua mesin virtual berhasil dibuat dan dikonfigurasi: VM 100 (Ubuntu Server) sebagai host layanan dan VM 101 (pfSense) sebagai gateway. Jaringan internal antara kedua VM terhubung melalui vmbr1 dengan konfigurasi IP statis, sementara koneksi ke internet dilakukan melalui vmbr0 yang mendapat IP dinamis dari router, sebagaimana ditunjukkan pada Gambar 8.

Copyright © 2026 The Author, Page 1858
This Journal is licensed under a Creative Commons Attribution 4.0 International License

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

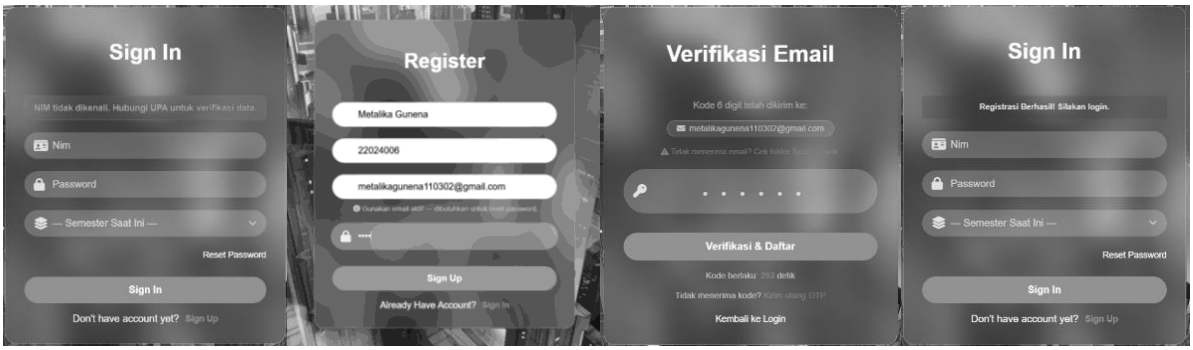


Gambar 8. Tampilan Dashboard Proxmox VE

Gambar 8 menunjukkan dashboard Proxmox VE yang menampilkan kedua mesin virtual yang telah dikonfigurasi (VM 100 dan VM 101) beserta status dan alokasi sumber dayanya, mengonfirmasi bahwa instalasi dan konfigurasi jaringan internal telah berjalan sesuai rancangan.

3.2 Portal Autentikasi

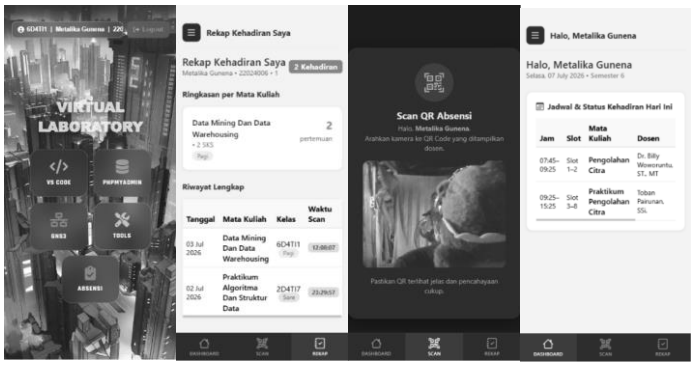
Portal autentikasi pada v-lab.my.id berhasil diimplementasikan dengan tiga halaman utama: halaman Sign In, halaman Register, dan halaman Verifikasi Email. Sistem validasi NIM berfungsi sesuai rancangan: mahasiswa yang NIM-nya tidak terdaftar dalam basis data tidak dapat menyelesaikan proses registrasi. Pengujian dilakukan menggunakan data mahasiswa kelas TI-1 dan berhasil memverifikasi seluruh mekanisme autentikasi. Ketiga halaman tersebut diimplementasikan dengan alur yang saling terhubung, mulai dari proses pendaftaran hingga verifikasi akun, sebagaimana ditunjukkan pada tampilan berikut di Gambar 9.



Gambar 9. Tampilan Portal Autentikasi Virtual Laboratory

Gambar 9 menunjukkan ketiga halaman utama portal autentikasi: halaman Sign In untuk pengguna yang sudah terdaftar, halaman Register untuk pendaftaran akun baru, dan halaman Verifikasi Email untuk konfirmasi kode OTP, yang secara keseluruhan membentuk alur autentikasi lengkap sesuai rancangan pada tahap Design.

Portal utama pada v-lab.my.id turut disediakan sebagai Progressive Web App (PWA), yaitu aplikasi web yang dapat dipasang (installable) langsung ke perangkat pengguna melalui manifest.json dan service worker (sw.js), tanpa melalui toko aplikasi. Setelah dipasang, portal dapat diakses seperti aplikasi native melalui ikon pada layar utama perangkat, sebagaimana ditunjukkan pada Gambar 10. Fitur ini bertujuan mempermudah akses cepat ke portal, khususnya bagi mahasiswa dan dosen yang perlu memindai kode QR pada modul Absensi menggunakan perangkat mobile.



Gambar 10. Tampilan Progressive Web App (PWA) Installable pada Portal Virtual Laboratory

Copyright © 2026 The Author, Page 1859
This Journal is licensed under a Creative Commons Attribution 4.0 International License

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

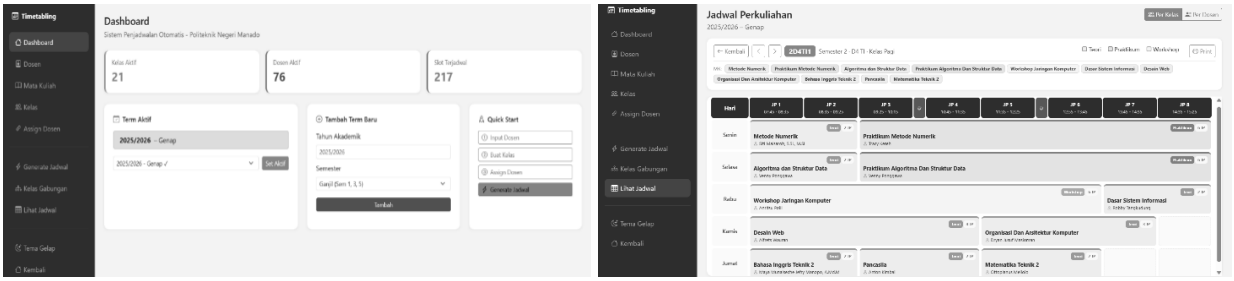
Gambar 10 menunjukkan tampilan portal setelah dipasang sebagai PWA, di mana ikon aplikasi muncul pada layar utama perangkat sebagaimana aplikasi native, memungkinkan akses cepat tanpa perlu membuka peramban dan mengetik alamat URL secara manual.

3.3 Layanan Laboratorium

Setelah berhasil login, mahasiswa dapat mengakses halaman beranda yang menampilkan menu layanan: VS Code, phpMyAdmin, GNS3, dan TOOLS (berisi shortcut ke PHP/HTML Runner dan File Manager). Seluruh layanan berhasil diakses melalui peramban web dari luar jaringan kampus. Beranda juga menyediakan tombol akses ke modul Absensi dan Timetable; admin dapat mengakses keduanya secara langsung, sedangkan dosen dan mahasiswa mengakses modul Absensi untuk melakukan presensi melalui pemindaian kode QR pada sesi yang dibuka oleh dosen, sebagaimana ditunjukkan pada Gambar 11.

Gambar 11. Tampilan Beranda Semua Role Virtual Laboratory

Gambar 11 menunjukkan tampilan beranda yang berbeda untuk masing-masing peran pengguna, dengan menu layanan ditampilkan dalam bentuk kartu yang dapat diakses langsung melalui satu klik, tanpa perlu mengingat alamat subdomain masing-masing layanan. Setelah pengguna mengakses layanan melalui menu beranda tersebut, khusus pengguna admin bisa mengakses modul Timetable, sistem akan menampilkan dashboard yang menyajikan ringkasan statistik penjadwalan berupa jumlah kelas aktif, dosen aktif, dan slot terjadwal, dilengkapi panel pengaturan term akademik serta menu quick start untuk mempercepat proses generate jadwal, seperti ditunjukkan pada Gambar 11.



Gambar 12. Tampilan Dashboard Timetable & Hasil Generate Jadwal Mata Kuliah

Gambar 12 menunjukkan hasil penjadwalan otomatis yang dihasilkan modul Timetable, diproses menggunakan algoritma greedy. Dashboard ini menampilkan ringkasan statistik jumlah dosen, mata kuliah, dan jadwal yang telah dibuat, pemilihan term akademik aktif beserta opsi cepat untuk membuat term baru atau menyalin jadwal dari term sebelumnya, serta tabel jadwal perkuliahan yang menunjukkan penempatan setiap mata kuliah pada hari dan slot waktu tertentu secara otomatis. Algoritma penjadwalan yang menghasilkan jadwal tersebut mengadopsi pendekatan greedy sebagai berikut: Topik penelitian makalah ini adalah menggunakan algoritma greedy untuk merancang sistem pemilihan dan penjadwalan mata kuliah yang lebih efisien [16]. Penerapan algoritma greedy untuk penjadwalan perkuliahan dan praktikum juga telah terbukti efektif pada penelitian-penelitian sejenis, di antaranya untuk menghasilkan jadwal perkuliahan tanpa bentrok yang menyesuaikan kapasitas kelas dan jumlah mahasiswa [17], serta untuk mengoptimalkan pembuatan jadwal praktikum laboratorium Teknik Informatika melalui algoritma greedy kombinasi dengan perulangan [18].

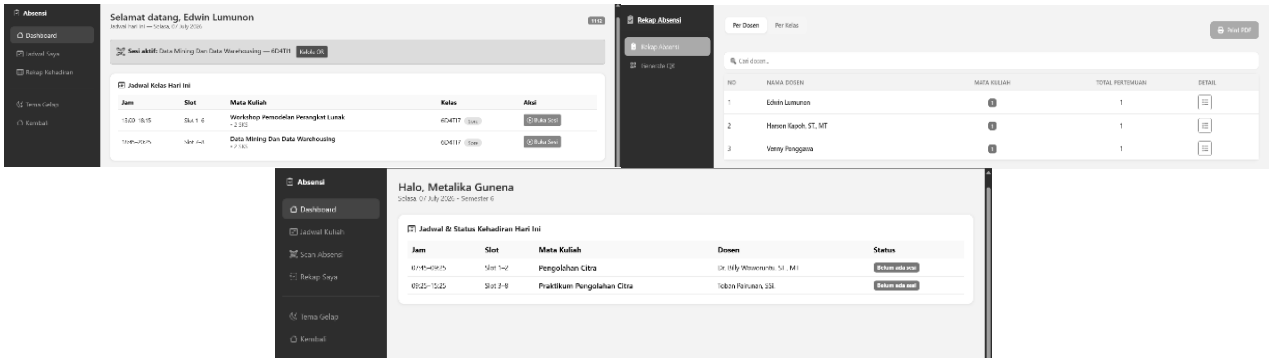
Secara formal, algoritma greedy menyelesaikan persoalan optimasi melalui lima elemen utama: himpunan kandidat, himpunan solusi, fungsi seleksi, fungsi kelayakan, dan fungsi obyektif. Pada modul Timetable, kandidat adalah seluruh pasangan mata kuliah dan kelas yang perlu dijadwalkan; solusi adalah subset yang berhasil ditempatkan ke slot jadwal; fungsi seleksi mengurutkan kandidat berdasarkan skor kritikalitas; fungsi kelayakan memeriksa bentrok kelas dan

Copyright © 2026 The Author, Page 1860
This Journal is licensed under a Creative Commons Attribution 4.0 International License

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

dosen melalui hash table; sedangkan fungsi obyektif memaksimalkan jumlah mata kuliah yang berhasil dijadwalkan tanpa bentrok. Karena algoritma ini tidak mengevaluasi seluruh kemungkinan solusi secara menyeluruh, hasilnya tidak selalu optimum global, sehingga diperlukan mekanisme fallback berupa saran penggabungan kelas lintas sesi Pagi/Sore ketika penempatan awal belum berhasil menjadwalkan seluruh mata kuliah [19].

Secara prosedural, algoritma tersebut dieksekusi melalui lima langkah berurutan pada modul Timetable: (1) seluruh pasangan mata kuliah dan kelas yang belum terjadwal dikumpulkan sebagai himpunan kandidat; (2) setiap kandidat diberi skor kritikalitas berdasarkan ketersediaan slot dan tingkat urgensi penjadwalan, kemudian diurutkan dari skor tertinggi ke terendah oleh fungsi seleksi; (3) kandidat dengan skor tertinggi diambil satu per satu, lalu fungsi kelayakan memeriksa potensi bentrok jadwal dosen dan kelas melalui pencarian hash table; (4) kandidat yang lolos pemeriksaan kelayakan ditempatkan ke slot jadwal dan dimasukkan ke himpunan solusi, sedangkan kandidat yang bentrok dicoba pada slot alternatif berikutnya sebelum ditandai untuk mekanisme fallback penggabungan kelas lintas sesi Pagi/Sore; dan (5) langkah kedua hingga keempat diulang untuk seluruh kandidat yang tersisa hingga himpunan kandidat kosong, dengan fungsi obyektif memaksimalkan jumlah mata kuliah yang berhasil ditempatkan pada setiap iterasi. Hasil penjadwalan yang dihasilkan oleh modul Timetable ini selanjutnya menjadi acuan bagi modul Absensi dalam menentukan jumlah pertemuan setiap mata kuliah, mengingat kehadiran mahasiswa dan dosen dihitung berdasarkan jumlah sesi perkuliahan yang telah dijadwalkan, sehingga kedua modul saling terhubung untuk menjaga konsistensi data akademik, sebagaimana ditunjukkan pada tampilan dashboard Absensi berikut pada Gambar 13.



Gambar 13. Tampilan Dashboard Absensi 3 Role

Gambar 13 menunjukkan tampilan dashboard modul Absensi pada peran dosen, yang menyajikan jadwal kelas hari ini beserta sesi aktif yang sedang berlangsung. Setiap sesi perkuliahan dilengkapi fitur Kelola QR untuk menghasilkan kode barcode presensi, sehingga mahasiswa dapat melakukan absensi dengan memindai kode tersebut melalui menu Scan Absensi pada dashboard masing-masing. Karena setiap sesi dibuat berdasarkan slot jadwal yang telah ditetapkan oleh modul Timetable, jumlah sesi absensi yang tersedia untuk satu mata kuliah otomatis mengikuti total pertemuan yang telah dijadwalkan, sehingga rekap kehadiran (baik per dosen maupun per mahasiswa) tetap konsisten dan tidak melebihi jumlah pertemuan yang ditentukan.

3.4 Pengujian Fungsional Layanan

Pengujian fungsional dilakukan untuk memverifikasi apakah setiap layanan laboratorium dapat diakses dan berjalan sesuai dengan yang dirancang. Hasil pengujian dirangkum pada Tabel 4.

Tabel 4. Hasil Pengujian Fungsional Layanan

Layanan	Skenario Uji	Hasil
Portal Login	Login dengan NIM & password valid	Berhasil
Registrasi	Daftar dengan NIM tidak terdaftar	Ditolak (sesuai)
OTP Email	Verifikasi kode OTP saat registrasi	Berhasil
Reset Password	Reset via OTP ke email terdaftar	Berhasil
VS Code	Membuka editor dan menulis kode	Berhasil
phpMyAdmin	Akses manajemen basis data	Berhasil
GNS3	Membuka antarmuka simulasi jaringan	Berhasil
PHP/HTML Runner	Menjalankan file .php dan .html	Berhasil
TinyFileManager	Upload/download file dari web	Berhasil
Redirect Rule	Akses subdomain tanpa login	Diarahkan ke login (sesuai)
Cloudflare Tunnel	Akses dari luar jaringan kampus	Berhasil
Absensi	Dosen membuka sesi QR; mahasiswa memindai untuk presensi	Berhasil
Timetable	Admin mengatur jadwal; dosen/mahasiswa melihat jadwal dengan penanda kelas gabungan	Berhasil

Copyright © 2026 The Author, Page 1861
This Journal is licensed under a Creative Commons Attribution 4.0 International License

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 5, August 2026 | Hal 1852-1864
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i5.1253

Sebagaimana dirangkum pada Tabel 4, seluruh tiga belas skenario pengujian fungsional berhasil dijalankan sesuai rancangan, mencakup autentikasi (login, registrasi, verifikasi OTP, reset password), seluruh layanan kontainer Docker (VS Code, phpMyAdmin, GNS3, PHP/HTML Runner, TinyFileManager), keamanan akses (Redirect Rule dan Cloudflare Tunnel), serta kedua modul pendukung operasional (Absensi dan Timetable). Dua di antaranya merupakan pengujian negatif yang sengaja dirancang untuk memverifikasi penolakan akses tidak sah: pendaftaran dengan NIM yang tidak terdaftar berhasil ditolak, dan upaya mengakses subdomain layanan tanpa login terlebih dahulu berhasil diarahkan kembali ke halaman login, yang menunjukkan bahwa mekanisme validasi dan proteksi akses berfungsi sesuai rancangan.

3.5 Efisiensi dan Keamanan Arsitektur Sistem

Pendekatan berbasis Docker container memberikan efisiensi lebih tinggi dibandingkan VM per mahasiswa yang diusulkan pada tahap awal, karena layanan dapat dibagikan ke seluruh mahasiswa dalam satu kelas tanpa duplikasi sistem operasi. Perlu dicatat bahwa hanya satu instans mesin virtual yang digunakan, meskipun banyak kontainer independen mungkin berjalan pada satu host Windows [20]. Virtualisasi berbasis kontainer merupakan alternatif dari virtualisasi menggunakan hypervisor, di mana kontainer berbagi seluruh sumber daya seperti perangkat keras, sistem operasi, dan pustaka pendukung sambil tetap menjaga abstraksi dan isolasi [12].

Cloudflare Tunnel dengan arsitektur Zero Trust terbukti efektif menggantikan kebutuhan IP publik statis: daemon cloudflared membuka koneksi keluar (outbound) sehingga alamat IP server tidak pernah terekspos dan permukaan serangan berbasis IP berkurang. Akses ke aplikasi tersebut melalui Cloudflare Tunnel yang menghubungkan server lokal ke jaringan Cloudflare, dan Cloudflare Access untuk mengelola autentikasi pengguna serta policy [13]. Cloudflare Zerotrust berperan penting dalam mengatasi keterbatasan IP publik, memungkinkan server diakses melalui domain pribadi meskipun layanan seperti FTP dan SSH hanya dapat dioperasikan di jaringan lokal [5]. Lapisan keamanan ini dilengkapi autentikasi berbasis NIM dan verifikasi OTP melalui email di sisi aplikasi. Modul Absensi yang membaca basis data Timetable secara langsung juga menjaga konsistensi data: perubahan jadwal atau penggabungan kelas otomatis tercermin pada dashboard Absensi tanpa sinkronisasi manual.

3.6 Keterbatasan Penelitian

Penelitian ini memiliki beberapa keterbatasan yang perlu diperhatikan: (1) pengujian dilakukan pada satu perangkat keras dengan spesifikasi terbatas sehingga belum dapat mengukur kapasitas pengguna simultan secara optimal; (2) layanan VS Code menggunakan ruang kerja bersama (shared workspace), sehingga isolasi data antar mahasiswa belum sepenuhnya diterapkan; (3) pengujian penerimaan pengguna (User Acceptance Test / UAT) belum dilaksanakan secara formal dan akan menjadi bagian dari penelitian lanjutan; (4) penyimpanan kata sandi pengguna masih menggunakan fungsi hash MD5 yang secara kriptografis tidak lagi direkomendasikan untuk data sensitif karena kerentanannya terhadap serangan collision dan rainbow table, sehingga migrasi ke algoritma hashing yang lebih aman seperti bcrypt atau Argon2 perlu menjadi prioritas pada pengembangan selanjutnya. penyimpanan kata sandi pengguna masih menggunakan fungsi hash MD5 yang secara kriptografis tidak lagi direkomendasikan untuk data sensitif karena kerentanannya terhadap serangan collision dan rainbow table, sehingga migrasi ke algoritma hashing yang lebih aman seperti bcrypt atau Argon2 perlu menjadi prioritas pada pengembangan selanjutnya [21].

3.7 Pembelajaran dari Penerapan NDLC

Penerapan metode NDLC dalam penelitian ini memberikan beberapa pembelajaran penting: (1) skalabilitas yang terencana, karena analisis matang di tahap awal memungkinkan kebutuhan sumber daya diprediksi sebelum sistem dibangun; (2) efisiensi biaya dan waktu, karena tahap simulasi pada laptop lokal mencegah kesalahan fatal saat implementasi di server produksi; (3) dokumentasi yang rapi, karena setiap tahapan NDLC menghasilkan dokumen yang krusial saat troubleshooting atau saat sistem diserahkan ke pengelola berikutnya; (4) iterasi berkelanjutan, karena sifat siklus NDLC memungkinkan peningkatan bertahap seperti penambahan kelas praktikum baru setelah migrasi ke server UPA TIK.

3.8 Pembahasan

Dibandingkan dengan penelitian Zulfikar, Bhawiyuga, dan Basuki yang mengimplementasikan laboratorium virtual multi klaster berbasis orkestrator Kubernetes dan mampu menangani 1000 pengguna konkuren pada utilisasi CPU 32,84% serta memori 77,60% [3], penelitian ini mengambil pendekatan yang berbeda: alih-alih berfokus pada orkestrasi kontainer lintas klaster antar-institusi, sistem yang dibangun mengintegrasikan langsung modul pendukung operasional laboratorium Absensi berbasis kode QR dan penjadwalan otomatis Timetable ke dalam satu portal yang sama, sebuah cakupan fungsional yang tidak menjadi fokus penelitian Zulfikar, Bhawiyuga, dan Basuki. Perbedaan fokus ini konsisten dengan Choudhary [1] dan Zhang & Guo [2], yang menekankan bahwa model laboratorium berbasis cloud/container umumnya dirancang untuk skalabilitas teknis, tanpa modul operasional akademik yang menyatu dengan proses pembelajaran sehari-hari.

Pada aspek layanan praktikum inti, penyediaan VS Code melalui kontainer Docker sejalan dengan pendekatan Malan yang menstandardisasi lingkungan pemrograman mahasiswa menggunakan kontainer agar konsisten lintas perangkat dan memudahkan pembaruan versi perangkat lunak di tengah semester [22]. Perbedaannya, penelitian Malan mengalokasikan satu kontainer terisolasi per mahasiswa melalui GitHub Codespaces untuk satu mata kuliah

Copyright © 2026 The Author, Page 1862
This Journal is licensed under a Creative Commons Attribution 4.0 International License

pemrograman, sedangkan penelitian ini menyediakan VS Code sebagai satu dari sembilan layanan yang diakses bersama dalam satu portal, sehingga isolasi data antar-mahasiswa pada layanan tersebut masih menjadi keterbatasan. Pada aspek simulasi jaringan, pendekatan ini sejalan dengan Morocho Román yang turut memanfaatkan Docker dan Cloudflare Tunnel untuk menyediakan akses jarak jauh terhadap laboratorium jaringan tanpa infrastruktur mahal [23]. Perbedaannya, Morocho Román mengintegrasikan Cisco Packet Tracer REST API dengan Google Colab untuk interaksi terprogram (otomasi jaringan) melalui Python, sedangkan penelitian ini menyediakan GNS3 sebagai layanan simulasi interaktif langsung melalui antarmuka grafis berbasis web tanpa memerlukan skrip otomasi tambahan.

Pada aspek penjadwalan, algoritma greedy yang diterapkan pada modul Timetable memiliki kemiripan prinsip dengan penelitian Akhmad, Khairuddin, dan Mhd. Zulfansyuri yang juga menerapkan algoritma greedy kombinasi dengan perulangan untuk penjadwalan praktikum pada skala satu program studi. Penelitian ini menangani kompleksitas tambahan berupa penjadwalan lintas kelas dan semester secara bersamaan, substitusi dosen pengampu saat terjadi benturan jadwal, serta mekanisme fallback berupa saran penggabungan kelas lintas sesi Pagi/Sore [18]. Di sisi lain, Coşar, Say, dan Dökeroğlu menunjukkan bahwa algoritma greedy dapat dioptimalkan lebih jauh dengan menerapkan hingga 120 heuristik berbeda dan diuji pada 21 studi kasus benchmark internasional (ITC-2007), mencapai nol pelanggaran batasan keras pada 18 di antaranya [24]. Penelitian ini belum melakukan pengujian benchmark sekomprehensif itu; algoritma yang diterapkan dioptimalkan secara spesifik untuk kebutuhan operasional Politeknik Negeri Manado, sehingga validitasnya masih terbatas pada konteks institusi ini.

Pada aspek presensi, modul Absensi berbasis kode QR memiliki tujuan serupa dengan sistem Asogwa, Onyedinma, Asogwa, dan Nkemdirim dan Bondoc, yaitu menggantikan pencatatan kehadiran manual dengan pemindaian kode QR. Sistem Asogwa, Onyedinma, Asogwa, dan Nkemdirim turut memungkinkan dosen mengelola jadwal kelas dan data kehadiran melalui satu antarmuka [25], sedangkan sistem Bondoc bergantung pada modul GSM terpisah untuk notifikasi SMS tanpa keterhubungan langsung dengan sistem penjadwalan [26]. Modul Absensi pada penelitian ini membaca basis data timetabling secara langsung, sehingga sesi presensi yang dibuka dosen otomatis mengacu pada jadwal kelas yang berlaku saat itu tanpa perlu konfigurasi sesi secara manual.

Secara keseluruhan, kontribusi penelitian ini bukan terletak pada satu algoritma atau teknik virtualisasi yang sepenuhnya baru, melainkan pada integrasi beberapa pendekatan yang masing-masing telah terbukti pada penelitian terdahulu virtualisasi berbasis container [1][2][3], penjadwalan berbasis greedy [18], [24], dan presensi berbasis QR [25], [26] ke dalam satu platform operasional yang saling terhubung untuk kebutuhan nyata satu institusi pendidikan.

4. KESIMPULAN

Penelitian ini menjalankan keseluruhan siklus Network Development Life Cycle, mulai dari identifikasi kesenjangan spesifikasi perangkat mahasiswa di Politeknik Negeri Manado pada tahap analisis, perumusan arsitektur berlapis yang memadukan Proxmox VE, pfSense, Docker, dan Cloudflare Tunnel pada tahap perancangan, verifikasi rancangan pada skala terbatas sebelum instalasi penuh, hingga penerapan sembilan layanan praktikum beserta modul pendukung operasional Timetable dan Absensi yang saling mengacu tanpa memerlukan sinkronisasi manual. Proses tersebut ditutup dengan tahap pemantauan beban infrastruktur dan penyusunan prosedur pengelolaan jangka panjang, sehingga sistem yang dihasilkan bukan sekadar purwarupa teknis, melainkan rancangan yang telah melalui pengujian menyeluruh dan siap dipertimbangkan untuk migrasi ke lingkungan produksi. Capaian utama penelitian ini terletak pada pembuktian bahwa ketiga model layanan cloud computing dapat berjalan bersamaan dalam satu portal akses tunggal tanpa memerlukan alamat IP publik statis, sekaligus menghadirkan otomatisasi operasional laboratorium yang selama ini masih dilakukan secara manual. Capaian tersebut sekaligus menjadi kontribusi utama penelitian ini, yaitu arsitektur Virtual Laboratory terintegrasi yang menggabungkan IaaS, PaaS, dan SaaS dalam satu portal, yang dapat dijadikan rujukan bagi pengembangan laboratorium praktikum berbasis cloud computing di lingkungan pendidikan tinggi lainnya. Dengan demikian, keterbatasan perangkat keras mahasiswa yang menjadi akar permasalahan di awal penelitian dapat diatasi melalui satu infrastruktur terpusat yang dikelola oleh institusi, bukan dibebankan pada kemampuan perangkat masing-masing individu. Sebagai kelanjutan, migrasi ke server UPA TIK, pengujian beban pengguna simultan, penguatan isolasi ruang kerja praktikum, serta pelaksanaan uji penerimaan pengguna secara formal menjadi langkah lanjutan yang diperlukan sebelum sistem ini diadopsi secara permanen dalam kegiatan akademik.

REFERENCES

[1] V. Choudhary, “Container-Based Virtual Labs for Scalable Cloud-Based Education,” *International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)*, vol. 1, no. 3, pp. 1–7, Oct. 2025, doi: 10.63345/v1.i3.101.

[2] Q. Zhang and H. Guo, “Investigation on Innovative Models of Computer Laboratories Based on Cloud Computing,” *Procedia Comput. Sci.*, vol. 228, pp. 383–390, Nov. 2023, doi: 10.1016/j.procs.2023.11.044.

[3] M. Zulfikar, A. Bhawiyuga, and A. Basuki, “Implementasi Lab Virtual berbasis Teknologi Kontainer Multi Klaster dengan Orkestrator Kubernetes,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 6, pp. 2649–2654, 2022, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11129

[4] D. Wahyuningsih, L. Affandi, and E. Sophia, “Evaluasi Performa Model Layanan Cloud Computing: Studi Komparatif Iaas, Paas, Dan Saas Dengan Pengujian Beban Bertingkat,” *Jurnal Teknologi Informasi, Teori, Konsep dan Implentasi (JTI-TKI)*, vol. 16, no. 2, pp. 94–101, Oct. 2025, doi: 10.36382/jti.

Copyright © 2026 The Author, Page 1863
This Journal is licensed under a Creative Commons Attribution 4.0 International License

[5] R. Abdur, F. Muhammad, and F. P. Ahmad, "Pemanfaatan Set Top Box Bekas Sebagai Server Cloud Hosting dengan Aapanel dan Cloudflare Zerotrust," *Blend Sains Jurnal Teknik*, vol. 4, no. 1, pp. 92–104, Jul. 2025, doi: 10.56211/blendsains.v4i1.985.
[6] P. Mell and T. Grance, "The NIST definition of cloud computing," Gaithersburg, MD, Sep. 2011. doi: 10.6028/NIST.SP.800-145.
[7] A. S. C. Polii, A. P. G. Manginsela, and R. Lumbu, *CLOUD COMPUTING*, 1st ed. Manado: Polimdo Press, 2024. [Online]. Available: http://press.polimdo.web.id
[8] F. H. Prasetyo, E. Infitharina, and F. Muhammad, "Penerapan Metode Network Development Life Cycle (NDLC) dalam Pengembangan Jaringan Komputer," *Journal of Informatics and Communications Technology (J-ICT)*, vol. 7, no. 1, pp. 80–087, Jun. 2025, doi: 10.52661.
[9] R. T. B. D. Butarbutar, G. M. A. Sasmita, and I. P. A. E. Pratama, "Development of a Notification-Based Network Security Monitoring System Using Network Development Life Cycle (NDLC)," *JITTER-Jurnal Ilmiah Teknologi dan Komputer*, vol. 4, no. 3, Dec. 2023, doi: 10.24843/JTRTI.2023.v04.i03.p01.
[10] Rahmadani, "Proxmox-Based Virtualization Techniques on Virtual Servers," *Proceedings of The International Conference on Computer Science, Engineering, Social Science, and Multi-Disciplinary Studies (CESSMUDS)*, vol. 1, 2025, doi: 10.64803/cessmuds.v1.68.
[11] L. Bitzki, D. Kreutz, and L. Bertholdo, "IoTEdu: Network Services for the Automated Management of Campus Internet-of-Things Networks," in *Anais da Escola Regional de Redes de Computadores (ERRC)*, Aug. 2025. doi: 10.5753/errc.2025.17739.
[12] S. Neelam *et al.*, "Load balancing and service discovery using Docker Swarm for microservice based big data applications," *Journal of Cloud Computing*, vol. 12, no. 1, Dec. 2023, doi: 10.1186/s13677-022-00358-7.
[13] S. Nandang and H. D. Muhammad, "Penerapan Keamanan Jaringan Zero Trust Network Access Berbasis Cloudflare Zero Trust Dalam Layanan Berbasis Internet Implementation Of Cloudflare Zero Trust Network Access Security In Internet-Based Services," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 8, no. 6, Nov. 2025, doi: 10.31539/sgyddq97.
[14] Y. Ariyanto, "Single Server-Side And Multiple Virtual Server-Side Architectures: Performance Analysis On Proxmox Ve For E-Learning Systems," *Journal of Engineering and Technology for Industrial Applications*, vol. 9, no. 44, pp. 25–34, Dec. 2023, doi: 10.5935/jetia.v9i44.903.
[15] M. Syamsu, V. Terisia, and U. Masduki, *Buku Ajar Jaringan Komputer Praktis Muda*. EUREKA MEDIA AKSARA, 2023. [Online]. Available: https://repository.penerbiteureka.com/publications/564522/buku-ajar-jaringan-komputer-praktis-mudah-disertai-studi-kasus
[16] J. Xiao, "The application of greedy algorithm in the course selection and scheduling system of college students," *Applied and Computational Engineering*, vol. 75, no. 1, pp. 48–53, Jul. 2024, doi: 10.54254/2755-2721/75/20240504.
[17] Y. M. Khader, Y. I. Nurhasanah, and A. D. Kartika, "Penjadwalan Matakuliah Menggunakan Algoritma Greedy (Studi Kasus Penjadwalan Semester Ganjil 2017-2018 Informatika Itenas)," *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 4, no. 3, pp. 207–213, Aug. 2018, doi: 10.33197/jitter.vol4.iss3.2018.168.
[18] R. Akhmad, N. Khairuddin, and S. Mhd. Zulfansyuri, "Implementasi Algoritma Greedy Kombinasi dengan Perulangan pada Aplikasi Penjadwalan Praktikum," *sudo Jurnal Teknik Informatika*, vol. 1, no. 2, pp. 42–50, Jun. 2022, doi: 10.56211/sudo.v1i2.8.
[19] E. S. Laela, W. Gata, and J. J. Purnama, "Optimalisasi Algoritma Greedy dalam Penyusunan Jadwal Pelajaran pada SMK Nurul Islam Cianjur," *Syntax Literate: Jurnal Ilmiah Indonesia*, vol. 7, no. 12, pp. 18451–18460, Dec. 2022, doi: 10.36418/syntax-literate.v7i12.10910.
[20] M. Sobieraj and D. Kotyński, "Docker Performance Evaluation across Operating Systems," *Applied Sciences (Switzerland)*, vol. 14, no. 15, pp. 1–16, Aug. 2024, doi: 10.3390/app14156672.
[21] A. R. Prasad, R. Chandramouli, A. Benslimane, and P. Langendorfer, *Cryptography and Network Security*, 1st ed. New York: River Publishers, 2022. [Online]. Available: www.riverpublishers.com
[22] D. J. Malan, "Containerizing CS50: Standardizing Students' Programming Environments," *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, vol. 1, pp. 534–540, Jul. 2024, doi: 10.1145/3649217.3653567.
[23] R. F. M. Román, "A cloud-accessible virtual lab architecture for teaching network automation: Integrating Cisco packet tracer REST API with docker, Cloudflare tunnel, and Google Colab," *Edelweiss Applied Science and Technology*, vol. 9, no. 9, pp. 144–154, Sep. 2025, doi: 10.55214/2576-8484.v9i9.9775.
[24] B. Coşar, B. Say, and T. Dökeroğlu, "A New Greedy Algorithm for the Curriculum-based Course Timetabling Problem," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 11, no. 2, pp. 1121–1136, Apr. 2023, doi: 10.29130/dubited.1113519.
[25] D. C. Asogwa, E. G. Onyedinma, E. C. Asogwa, and E. U. Nkemdirim, "Quick Response (QR) Code Based Students' Attendance System in the University," *International Journal of Research and Innovation in Applied Science (IJRIAS)*, vol. 10, no. 7, pp. 1–6, Jul. 2025, doi: 10.51584/IJRIAS.2025.100700142.
[26] B. C. Bondoc, "Web-based smart attendance monitoring system using QR code and GSM module for SMS notificatio," *World Journal of Advanced Research and Reviews*, vol. 20, no. 3, pp. 1843–1849, Dec. 2023, doi: 10.30574/wjarr.

Copyright © 2026 The Author, Page 1864
This Journal is licensed under a Creative Commons Attribution 4.0 International License