

Sistem *Scheduling* Pelaporan Data Akademik di UIN Sunan Kalijaga ke Pangkalan Data Pendidikan Tinggi (PDDikti) dengan Menggunakan Fitur *Cron Job* di Linux

Adi Wirawan

PTIPD UIN Sunan Kalijaga, Yogyakarta

e-mail : adi.wirawan@uin-suka.ac.id.

Artikel ini diajukan 3 Mei 2020, direvisi 12 Juni 2020, diterima 12 Juni 2020, dan dipublikasikan 9 November 2020.

Abstract

Data reporting from higher education to the Higher Education Data Centre (PDDikti) is a form of reporting that must be sent to The Indonesian Ministry of Research, Technology and Higher Education (Ristekdikti). This reporting includes aspects of students, academic transactions, and some other data, which usually has a time limit for each academic year. UIN Sunan Kalijaga like other higher educutions, reports a lot of data to PDDikti with a limited time, which sometimes makes reporting officers need to do overtime to report existing data. This study aims to create an application that can schedule data reporting times to the PDDikti feeder application, to reduce overtime that needs to be done by the operator. The application is created using the cron job feature owned by Linux to run reporting commands.

Keywords: *Cron Job, PDDikti, Academic Data*

Abstrak

Pelaporan data dari perguruan tinggi ke Pangkalan Data Pendidikan Tinggi (PDDikti) merupakan bentuk pelaporan yang wajib dikirimkan ke Kementerian Ristek Teknologi dan Pendidikan Tinggi (Kemenristek Dikti). Pelaporan ini meliputi aspek mahasiswa, transaksi akademik, dan beberapa data yang lainnya, yang biasanya memiliki batasan waktu pada tiap tahun akademiknya. UIN Sunan Kalijaga seperti perguruan tinggi lainnya, melaporkan banyak data ke PDDikti dengan tenggat waktu yang sudah ditentukan, yang kadang menjadikan petugas pelaporan perlu melakukan lembur untuk melaporkan data yang ada. Penelitian ini bertujuan untuk membuat aplikasi yang bisa menjadwalkan waktu pelaporan data ke aplikasi *feeder* PDDikti, sehingga bisa mengurangi waktu lembur yang perlu dilakukan oleh operator. Aplikasi dibuat dengan memanfaatkan fitur *cron job* yang dimiliki oleh Linux untuk menjalankan perintah pelaporan.

Kata Kunci: *Cron Job, PDDikti, Data Akademik*

1. PENDAHULUAN

Pelaporan data akademik ke Pangkalan Data Pendidikan Tinggi atau PDDikti adalah hal yang sangat penting untuk dilakukan. Menurut situs dari PDDikti (PDDikti, 2020), PDDikti merupakan kumpulan data penyelenggaraan pendidikan tinggi seluruh perguruan tinggi yang terintegrasi secara nasional. Pelaporan tersebut mengirimkan banyak informasi ke Pangkalan Data Pendidikan Tinggi yang dikelola oleh Ristek Dikti. Data yang dikirimkan ke PDDikti adalah data transaksi akademik yang ada di institusi pendidikan, salah satunya UIN Sunan Kalijaga Yogyakarta. Data yang dikirimkan diantaranya adalah data aktifitas mengajar dosen, data aktifitas perkuliahan mahasiswa, data matakuliah, data kurikulum, data penyelenggaraan matakuliah, dan data-data akademik lainnya.

Dengan begitu pentingnya data yang perlu masuk ke PDDikti, Ristek Dikti menjadikan data yang terkandung di PDDikti ini menjadi sumber untuk valid atau tidaknya profil seorang mahasiswa. Tidak jarang seorang lulusan dari suatu institusi pendidikan menjadi tidak diterima bekerja karena tidak tersedianya data lulusan di situs PDDikti.

Selama ini pelaporan ke PDDikti di UIN Sunan Kalijaga biasa dikerjakan oleh pihak Akademik. Pelaporan dilakukan ketika jam kerja untuk melaporkan banyak data yang harus segera masuk.



Penting untuk diketahui, bahwa periode untuk memasukkan transaksi bisa jadi dibatasi oleh pihak Ristek Dikti, sehingga pihak Akademik UIN Sunan Kalijaga melakukan lembur untuk menyelesaikan proses pelaporan. Dengan terbatasnya waktu jam kerja maka proses pelaporan bisa mengalami kendala waktu untuk menyelesaikan banyaknya laporan yang harus dikirimkan.

Server UIN Sunan Kalijaga secara umum tidak mengalami beban pekerjaan yang berat pada saat malam hari, karena pada malam hari tidak banyak yang melakukan transaksi ke *server*. Beban *server* yang besar biasa diterima pada saat jam kerja atau pada saat pengisian KRS (Kartu Rencana Studi). Malam hari merupakan waktu di mana *resource* ataupun sumber daya yang dimiliki *server* tidak digunakan dengan maksimal. *Resource* tersebut bisa dimanfaatkan untuk mengerjakan laporan-laporan yang membutuhkan banyak sumber daya untuk menjalankan proses yang dibutuhkan. Terbatasnya waktu operator dalam mengerjakan laporan pada saat malam hari tentu menjadi pertimbangan untuk memberikan tugas tambahan kepada operator terutama operator PDDikti untuk menjalankan tugas pada malam hari.

Cron job merupakan aturan penjadwalan suatu perintah yang dijalankan secara periodik (Setiawan, 2013). *Cron* diatur oleh *crontab (cron table) file*, yakni adalah sebuah *file* konfigurasi yang berisikan perintah yang akan berjalan secara periodik berdasarkan pengaturan jadwal yang diberikan. Kontrol untuk mengatur waktu akan diatur dengan menggunakan PHP (*Hypertext Preprocessors*) sebagai bahasa pemrograman, di mana PHP adalah bahasa yang kita gunakan untuk membuat *server* menghasilkan *output* yang berbeda setiap kali *browser* melakukan *request* terhadap suatu halaman (Nixon, 2012). Penelitian ini bertujuan untuk membuat program penjadwalan pelaporan data Akademik di UIN Sunan Kalijaga ke *server* PDDikti yang dikelola oleh Ristek Dikti. Penelitian ini fokus kepada pemanfaatan fitur *cron job* yang dimiliki oleh Linux untuk menjadwalkan pelaporan pada waktu yang bisa dijadwalkan oleh seorang operator.

Penelitian ini memanfaatkan fasilitas *cron job* di Linux untuk menjadwalkan proses eksekusi program. Pemanfaatan *cron job* untuk melakukan proses penjadwalan telah banyak digunakan di beberapa penelitian di antaranya seperti pengambilan data secara periodik di Arduino (Isikdag & Pilouk, 2016) pengendalian rumah pintar (Risaldo et al., 2016), monitoring kualitas air di PDAM Surabaya (Budiarti et al., 2017), pengambilan data temperatur (Ziyaulhaq, 2017), pembuatan bel bergantian otomatis (Moedjahedy, 2018), proses menampilkan jadwal matakuliah dalam *signate board* atau papan informasi (Triyono, 2018).

Beberapa penelitian sebelumnya telah memanfaatkan *cron job* di Linux untuk eksekusi beberapa program secara periodik, baik di sistem Arduino, rumah pintar, dan beberapa program yang lainnya. Penelitian ini difokuskan untuk memanfaatkan *cron job* di Linux untuk mengirimkan laporan data akademik ke PDDikti, berdasarkan data dan waktu yang bisa ditentukan oleh operator pelaporan data PDDikti.

Tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut:

- 1) Membangun sistem yang bisa mengatur penjadwalan proses pelaporan data ke PDDikti.
- 2) Mengimplementasikan sistem pelaporan dengan sistem penjadwalan yang bisa memberikan *log* hasil pelaporan.
- 3) Mengimplementasikan fitur *cron job* yang ada di Linux untuk menjalankan perintah secara dalam periode waktu tertentu.

2. METODE PENELITIAN

Pengembangan sistem pelaporan data ke PDDikti dengan menggunakan fasilitas *cron job* yang ada di Linux ini dimulai dengan melakukan studi pendahuluan. Tahapan studi pendahuluan ditujukan untuk mengetahui konsep yang mendalam tentang hal-hal yang dibutuhkan dalam penelitian. Tahapan ini dilakukan untuk melakukan proses pengkajian data atau informasi yang dibutuhkan dalam proses pengembangan sistem.



- 1) Studi Literatur dan Kepustakaan
Studi literatur atau kepustakaan adalah jenis studi data yang dilakukan dengan cara menelaah buku-buku ataupun literatur-literatur yang bisa dimanfaatkan sebagai landasan serta teori dalam pembuatan sistem.
- 2) Observasi
Merupakan studi yang dilakukan dengan cara melihat secara langsung tempat ataupun objek untuk bisa mendapatkan informasi yang diperlukan dalam pembuatan sistem.
- 3) Wawancara
Tahapan yang dilakukan dengan mengajukan pertanyaan kepada pihak terkait untuk mendapatkan informasi untuk bisa mengetahui masalah yang sedang dihadapi.

Metodologi yang digunakan dalam penelitian ini adalah jenis metodologi pengembangan sistem (SDLC) *waterfall*. Metode *waterfall* adalah metode yang memecah tahapan proses pengerjaan proses menjadi urutan yang linier, di mana pengerjaan suatu proses tergantung dari proses sebelumnya (Kontributor Wikipedia, 2020). Analisis kebutuhan sistem pelaporan ini terdiri dari analisis kebutuhan fungsional dan kebutuhan non fungsional.

2.1. Analisis Kebutuhan Fungsional Sistem

Kebutuhan fungsional dari sistem pelaporan ini adalah:

- 1) Sistem pelaporan bisa menyimpan jadwal waktu kapan suatu pelaporan akan dilakukan.
- 2) Sistem bisa menyimpan *log* hasil pelaporan sebagai bentuk koreksi semisal ada hal yang perlu diperbaiki dalam proses pelaporan.
- 3) Sistem bisa menampilkan daftar pelaporan yang sudah berlangsung, sedang berlangsung ataupun yang akan berlangsung.

2.2. Analisis Kebutuhan Non Fungsional Sistem

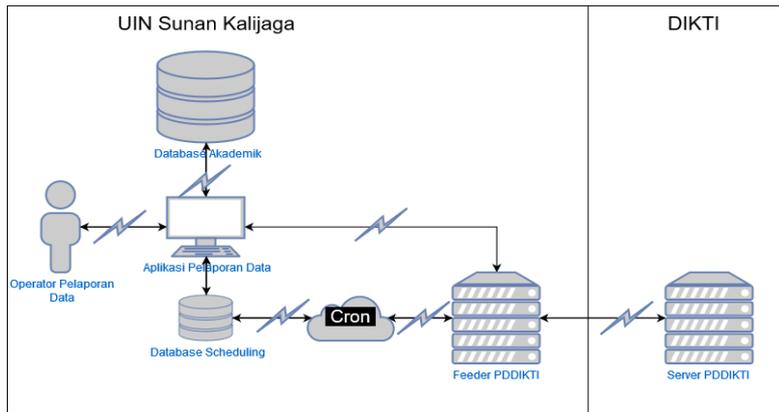
Analisis kebutuhan non fungsional memberikan gambaran tentang kebutuhan sistem yang memiliki titik berat kepada properti perilaku yang dimiliki oleh suatu sistem yang mendukung kelancaran proses pembuatan dan pengujian. Beberapa aspek yang dibutuhkan dalam pembuatan sistem ini di antara kebutuhan perangkat keras (*hardware*), perangkat lunak (*software*), dan pengguna (*user*) sebagai bahan analisis kebutuhan yang harus dipenuhi dalam proses perancangan sistem yang akan diterapkan.

- a) Analisis Kebutuhan Perangkat Keras (*Hardware*)
Kebutuhan *hardware* yang diperlukan pada proses pembangunan dan implementasi terdiri dari:
 - Server processor Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz. Memory 200GB
 - Laptop processor Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00GHz. Memory 8GB
- b) Analisis Kebutuhan Perangkat Lunak (*Software*)
Kebutuhan *software* yang diperlukan pada proses pembangunan dan implementasi terdiri dari:
 - PHP versi 5.5.9
 - Database PostgreSQL 12.1
 - Ubuntu Server 4.04
 - Apache Web Server 2.4.7
 - Visual Studio Code 1.45.0
- c) Analisis Pengguna (*User*)
Tahapan ini untuk mengetahui siapa saja aktor yang terlibat dalam menjalankan sistem. Pengguna sistem adalah operator di bagian Akademik yang akan melaporkan data-data Akademik ke PDDikti.

2.3. Desain Pengembangan Sistem

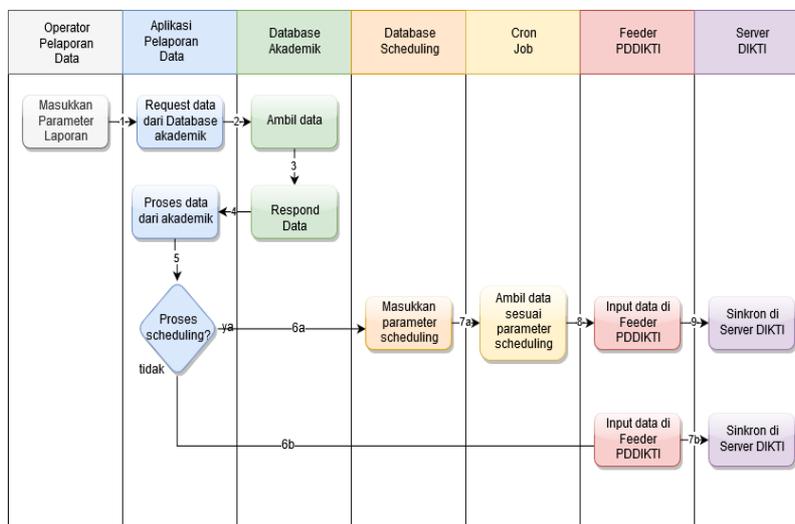
Rancangan sistem pelaporan data ke *feder* PDDikti dengan menggunakan *cron job* di Linux secara keseluruhan dapat dilihat pada Gambar 1.





Gambar 1. Skema pelaporan data ke feeder PDDikti

Alur proses pelaporan data ke PDDikti dengan menggunakan proses *scheduling* atau penjadwalan ditampilkan pada Gambar 2.



Gambar 2. Alur proses pelaporan data PDDikti dengan *scheduling*

Alur proses pelaporan data ke PDDikti dengan menggunakan *scheduling* ditampilkan pada Gambar 2. Alur yang terjadi pada Gambar 2 adalah:

- 1) Operator memasukkan parameter yang hendak dilaporkan ke dalam sistem pelaporan PDDikti yang ada di UIN Sunan Kalijaga.
- 2) Aplikasi pelaporan melakukan *request* data ke *database* akademik untuk mendapatkan data sesuai dengan operasi yang akan dilakukan oleh operator pelaporan data.
- 3) *Database* akademik memberikan *respond* berupa data yang dipanggil oleh aplikasi melalui *backend* yang dikembalikan dengan *protocol* REST.
- 4) Data yang diperoleh dari *database* akademik melalui *backend* kemudian diterima oleh aplikasi. Data kemudian diolah dan diformat sesuai dengan format yang sudah ditentukan oleh *feeder* PDDikti.
- 5) Aplikasi menyiapkan dua pilihan pada saat akan melakukan pelaporan. Pelaporan dengan menggunakan proses *scheduling* atau penjadwalan ataukah langsung melakukan insert data ke *feeder* PDDikti.
- 6) Jika dipilih perintah untuk proses *scheduling* maka aplikasi akan memasukkan data parameter ke *database scheduling* melalui *backend*; Jika dipilih proses input data tanpa menggunakan proses *scheduling* maka aplikasi akan melanjutkan proses memasukkan data ke *feeder* PDDikti.



- 7) Data yang ada di *database scheduling* secara periodik akan diambil datanya oleh *cron job* di Linux. Apabila waktu yang ditentukan saat melakukan penjadwalan sesuai, atau lebih dari waktu yang ditentukan, serta belum diproses maka parameter yang dimasukkan akan diproses, dan data akan dimasukkan ke *feeder PDDikti*; Proses sinkronisasi data di *feeder PDDikti* dengan server DIKTI.
- 8) Parameter yang diatur pada saat *scheduling* akan dimasukkan ke *feeder PDDikti*.
- 9) Data yang ada di *feeder PDDikti* disinkronkan dengan data yang ada di server DIKTI.

3. HASIL DAN PEMBAHASAN

Implementasi sistem *scheduling* pelaporan data akademik di UIN Sunan Kalijaga ke Pangkalan Data Pendidikan Tinggi (PDDikti) dengan menggunakan fitur *cron job* di Linux menggunakan bahasa pemrograman PHP, HTML, *database PostgreSQL* dan fitur *cron job* di linux. PHP digunakan sebagai bahasa pemrograman yang digunakan untuk melakukan proses input, update, delete data, serta proses manipulasi data. HTML digunakan sebagai antar muka pengguna untuk memasukkan jadwal pelaporan data.

Database PostgreSQL yang digunakan pada aplikasi ini merupakan sistem pengaturan *database* relasional yang awalnya bermula di University of California (Obe & Hsu, 2015). *PostgreSQL* digunakan untuk menyimpan jadwal pelaporan data serta menyimpan *log* hasil pelaporan. *Cron job* di Linux digunakan untuk menjalankan proses pengecekan jadwal yang disimpan di *database*, jika jadwal yang dimasukkan sesuai dengan waktu berjalannya *cron job* atau lebih dari waktu berjalannya *cron* dan tidak dalam status diproses maka proses akan dijalankan.

Pembuatan skema tabel yang diterapkan di *database PostgreSQL* dimaksudkan sebagai lokasi penyimpanan untuk menyimpan jadwal pelaporan data serta untuk menyimpan *log* dari hasil pelaporan data.

Tabel 1. Skema Tabel Penjadwalan dan Penyimpanan Log.

Nama Kolom	Jenis Kolom
Id	Int
Url	Text
Log	Text
tgl_cron	Timestamp
Nip	Varchar(20)
Status	Varchar(10)
tgl_cron_akses	Timestamp
tgl_set	Timestamp
Page	Varchar(20)

Jadwal pelaporan data serta parameter yang tersimpan di *database PostgreSQL* akan dieksekusi oleh layanan yang disediakan oleh Linux yang bernama *cron job*. *Cron job* di Linux digunakan untuk mengatur interval waktu yang diinginkan untuk proses pemanggilan perintah di Linux. Perintah di Linux ini nantinya akan diisi dengan sebuah alamat *website* yang berisikan perintah untuk mengambil data-data yang diperlukan untuk dimasukkan ke server *feeder PDDikti*.

```
* /5 * * * * wget -q -O /dev/null "http://alamat-website" > /dev/null 2>&1
```

Gambar 3. Perintah *cron job* untuk menjalankan pemanggilan program.

Perintah yang dijalankan di *cron job* di Gambar 3 adalah perintah untuk menjalankan perintah *wget* di Linux yakni memanggil alamat sebuah *website*. Angka yang dituliskan sebelum perintah



di *cron job* menandakan waktu eksekusi perintah (Siever et al., 2009). Pada proses pelaporan, sistem menggunakan script PHP melakukan proses pelaporan data. Antar muka pelaporan data dibuat dengan menggunakan HTML sebagai antarmukanya. PHP sebagai bahasa pemrograman untuk pemrosesan data juga untuk proses pelaporan data.

Gambar 4 menampilkan *form* yang digunakan untuk memasukkan data-data yang diinginkan untuk dimasukkan ke aplikasi *feeder* PDDikti. *Form* yang tampil di Gambar 4, merupakan salah satu pelaporan data mahasiswa yang harus dimasukkan ke aplikasi *feeder* PDDikti. Aplikasi menyediakan beberapa pilihan fakultas, prodi, angkatan dan beberapa pilihan lainnya. Pengaturan waktu *cron* diatur di area atur jadwal.

Area atur jadwal *cron* menyediakan inputan waktu, yang mana saat waktu yang sudah diatur sesuai atau melebihi waktu yang ditentukan dan status pelaporan belum diproses maka proses input laporan ke *feeder* PDDikti akan dijalankan. Ada beberapa halaman pelaporan yang disiapkan untuk proses pelaporan ini, salah satunya adalah halaman pelaporan data mahasiswa seperti yang tampil di Gambar 4.

Web Service Mahasiswa

Halaman ini ditujukan untuk mengisi data peserta didik pada feeder.

Nama Fakultas	<input type="text" value="Pascasarjana"/>
Nama Prodi	<input type="text" value="-Semua Prodi-"/>
NIM (optional)	<input type="text"/>
Masukkan NIM yang anda ingin masukkan. Bisa dipisahkan dengan tanda koma (,) antar NIM.	
Rentang Angkatan	Batas Bawah <input type="text"/> Batas Atas <input type="text"/>
(Kosongkan rentang Angkatan jika ingin semua angkatan. Lama.)	
Rentang Record	Batas Bawah <input type="text"/> Batas Atas <input type="text"/>
(Kosongkan rentang Record jika ingin semua record. Lama.)	
Atur Jadwal Cron	<input type="text" value="26-03-2020 00:00:00"/> <input type="button" value="Set Cron"/>

Gambar 4. Antar Muka Pelaporan Data.

Hasil eksekusi yang dijadikan sebagai pengujian aplikasi dilakukan dengan menggunakan layanan yang disediakan oleh aplikasi, menjadwalkan waktu pelaporan, serta melihat *log* yang didapatkan ketika proses pelaporan sudah terjadi. Gambar 5 menampilkan halaman khusus yang bisa digunakan untuk mengetahui jadwal laporan yang akan dijalankan, sudah dijalankan, maupun yang dalam proses berjalan. Dalam antarmuka yang ditampilkan di Gambar 5, ada pilihan yang bisa dipilih oleh pengguna aplikasi yakni, fakultas, prodi, status *cron*, dan juga halaman yang merupakan halaman yang dilaporkan ke *feeder* PDDikti.

Disediakan kolom waktu *cron* di Gambar 5 untuk bisa mengetahui kapan suatu jadwal pelaporan akan dijalankan. Kolom selanjutnya, yakni kolom waktu atur adalah kolom yang berisikan waktu suatu perintah laporan diatur oleh admin pengelola laporan. Kolom detail, merupakan kolom yang berisikan tombol yang apabila diklik akan menampilkan Gambar 6 yang merupakan hasil *output* ketika proses pelaporan berjalan. *Output* pelaporan bisa berupa informasi sukses, ataupun juga bisa berisikan informasi gagal terhadap laporan yang dijalankan. Dengan adanya fitur 6 ini diharapkan operator pelaporan ketika laporan sudah selesai, maka operator bisa melakukan koreksi apakah pelaporan yang dilakukan berhasil ataukah gagal.



Data Log Cron

Halaman ini ditujukan untuk mencatat Log Cron.

Nama Fakultas:

Nama Prodi:

Halaman:

Status Cron:

NO	Fakultas	Halaman	Status	Waktu Cron	Waktu Atur	Detail	Aksi
1	Adab dan Ilmu Budaya	Web Service Nilai	Selesai	03/03/2020 01:32:57	02/03/2020 15:59:18	Cek	Del
2	Adab dan Ilmu Budaya	Web Service Nilai	Selesai	03/03/2020 00:17:26	02/03/2020 15:59:01	Cek	Del
3	Ilmu Tarbiyah dan Keguruan	Web Service Kelas Mata Kuliah	Selesai	02/03/2020 22:09:55	02/03/2020 15:56:50	Cek	Del

Gambar 5. Tampilan antar muka daftar pelaporan.

Info

Halaman : Web Service Nilai

Fakultas : Adab dan Ilmu Budaya

Prodi : Sejarah Peradaban Islam (S2)

Tahun / Semester : 2019 / 2

Angkatan : 2015 s.d. 2019

Rentang Baris : 1 s.d. 400

Log : PROSES untuk 177 data telah selesaiBerhasil 172, belum berhasil 5
insertNIM 19201022016 pada (S2SKI16#USK501005) Filsafat Ilmu dengan nilai (0)
insert Sukses Dimasukkan ke nilai.
insertNIM 19201020015 pada (S2SKI16#USK501005) Filsafat Ilmu dengan nilai (0)
insert Sukses Dimasukkan ke nilai.
insertNIM 19201022001 pada (S2SKI16#USK501005) Filsafat Ilmu dengan nilai (0)

Gambar 6. Log pelaporan data saat cron job dijalankan

Berdasarkan hasil pengujian yang dilakukan dengan cara melihat hasil *log* dari aplikasi, maka bisa dilihat di Gambar 6 bahwa aplikasi sudah berhasil melakukan proses input data ke server *feeder* PDDikti dengan mengembalikan informasi sukses.

4. KESIMPULAN

Berdasarkan dari hasil penelitian dan pembahasan yang dilakukan maka diperoleh kesimpulan sebagai berikut:

- 1) Penelitian ini berhasil membangun sistem yang mampu menyimpan jadwal waktu kapan suatu pelaporan akan dilakukan.
- 2) Penelitian ini berhasil membangun sistem yang bisa menyimpan *log* hasil pelaporan untuk koreksi.
- 3) Penelitian ini bisa membangun sistem yang menampilkan daftar pelaporan yang sudah berlangsung, sedang berlangsung ataupun yang akan berlangsung.



DAFTAR PUSTAKA

- Budiarti, R. P. N., Widyatmoko, N., Hariadi, M., & Purnomo, M. H. (2017). Web scraping for automated water quality monitoring system: A case study of PDAM Surabaya. *Proceeding - 2016 International Seminar on Intelligent Technology and Its Application, ISITIA 2016: Recent Trends in Intelligent Computational Technologies for Sustainable Energy*. <https://doi.org/10.1109/ISITIA.2016.7828735>
- Isikdag, U., & Pilouk, M. (2016). Integration of GEO-sensor feeds and event consumer services for real-time representation of IoT nodes. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. <https://doi.org/10.5194/isprsarchives-XLI-B4-267-2016>
- Kontributor Wikipedia. (2020). *Model waterfall*. Wikipedia, Ensiklopedia Bebas. https://id.wikipedia.org/wiki/Model_waterfall
- Moedjahedy, J. (2018). Implementasi Cron Job Linux Sebagai Bel Pergantian Kelas Otomatis Di Universitas Klatat. *CogITo Smart Journal*. <https://doi.org/10.31154/cogito.v4i1.97.1-10>
- Nixon, R. (2012). Learning PHP, MySQL, JavaScript, and CSS, 2nd Edition. In *O'Reilly Media*.
- Obe, R., & Hsu, L. (2015). PostgreSQL: Up and Running. In *O'Reilly Media*.
- PDDikti. (2020). *Sejarah PD Dikti*. PDDikti - Pangkalan Data Pendidikan Tinggi. https://pddikti.kemdikbud.go.id/sejarah_pd_dikti
- Risaldo, B., Putra, A., Rohadi, E., & Ariyanto, R. (2016). PENGENDALIAN RUMAH PINTAR MENGGUNAKAN JARINGAN INTERNET BERBASIS RASPBERRY PI. *Prosiding SENTIA*, 103–110.
- Setiawan, A. (2013). *RANCANG BANGUN SISTEM MONITORING RUANGAN MENGGUNAKAN WEBCAM BERBASIS OPENWRT*. UIN Sunan Kalijaga.
- Siever, E., Figgins, S., Love, R., & Robbins, A. (2009). Linux in a nutshell. In *O'Reilly*.
- Triyono, A. (2018). IMPLEMENTASI RASPBERRY PI UNTUK APLIKASI SIGNATE BOARD JADWAL KULIAH BERBASIS WEB DENGAN SISTEM OPERASI LINUX. *Sebatik*, 22(2), 102–106. <https://doi.org/10.46984/sebatik.v22i2.315>
- Ziyaulhaq, M. (2017). Embedded Web server for Industrial Applications using Raspberry-Pi. In *International Journal of Engineering Science and Computing*.

