
WEB-BASED ACADEMIC INFORMATION SYSTEM OF YAPI SUNNI TEGALGONDO ISLAMIC BOARDING SCHOOL USING WATERFALL METHOD TO MANAGE STUDENTS ACADEMIC DATA

Sang Aji Indutoro^{*1}, Endah Sudarmilah²

^{1,2}Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta, Indonesia
Email: ¹1200180003@student.ums.ac.id , ²endah.sudarmilah@ums.ac.id

(Naskah masuk: 26 Januari 2022, Revisi : 29 Januari 2022, diterbitkan: 25 Februari 2022)

Abstract

Yapi Sunni Islamic Boarding School is one of the Islamic da'wah and education foundations located in Tegalondo, Wonosari, Klaten. Yapi is active in the management of religious education with the vision of creating a generation of Muslims who are expected to be the successors of future generations who are tough and knowledgeable. The students' academic data management system to report generation, Yapi Islamic Boarding School uses a combination of manual data recording with paper media, archives, and Microsoft Office which in its application often problems arise when collecting data manually and the possibility of being damaged, lost or exposed to a virus attack is quite high. This study aims to design a web-based Islamic boarding school academic information system that can provide access for aliyah level student academic data managers and provide access for student parents to monitor their child's development. The system will be designed using the PHP programming language, MySQL as database management, and the CodeIgniter framework. The system development method used in this research is the waterfall method. The result of this research is a web-based Yapi Sunni Islamic boarding school information system. Based on the results of blackbox testing, this system can run well and the results of UAT testing carried out by users, obtained an average percentage of 93%, it can be concluded that the system is in accordance with the needs and can be accepted by user.

Keywords: *academic information system, codeigniter, islamic boarding school, waterfall, web.*

SISTEM INFORMASI AKADEMIK PONDOK PESANTREN YAPI SUNNI TEGALGONDO BERBASIS WEB MENGGUNAKAN METODE WATERFALL UNTUK MENGEOLA DATA AKADEMIK SANTRI

Abstrak

Pondok pesantren Yapi Sunni merupakan salah satu yayasan dakwah dan pendidikan islam yang terletak di desa Tegalondo, Wonosari, Klaten. Yapi aktif dalam pengelolaan pendidikan keagamaan dengan visi mencetak generasi Islam yang diharapkan mampu menjadi penerus generasi masa depan yang tangguh dan berilmu. Sistem pengelolaan data akademik santri hingga pembuatan laporan, pondok pesantren Yapi menggunakan perpaduan antara pencatatan data secara manual dengan media kertas, arsip, dan *Microsoft Office* yang dalam penerapannya sering muncul kendala saat melakukan pendataan secara manual serta kemungkinan untuk rusak, hilang atau terkena serangan virus cukup tinggi. Penelitian ini bertujuan untuk rancang bangun sistem informasi akademik pondok pesantren berbasis web yang dapat memberi akses bagi pengelola data akademik santri tingkat aliyah dan menyediakan akses untuk wali santri dalam memantau perkembangan anak mereka. Sistem dirancang menggunakan bahasa pemrograman *PHP*, *MySQL* sebagai manajemen databasenya, dan *framework CodeIgniter*. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *waterfall*. Hasil dari penelitian ini berupa sistem informasi akademik pondok pesantren Yapi Sunni berbasis web. Berdasarkan hasil pengujian *blackbox*, sistem ini dapat berjalan dengan baik dan hasil pengujian *UAT* yang dilakukan oleh pengguna, diperoleh rata-rata presentase sebesar 93%, maka dapat disimpulkan sistem sesuai dengan kebutuhan dan dapat diterima oleh pengguna.

Kata kunci: *codeigniter, pondok pesantren, sistem informasi akademik, waterfall, web*

1. PENDAHULUAN

Perkembangan teknologi saat ini banyak memberikan dampak positif bagi manusia [1]. Implementasi sistem informasi merupakan salah satu bentuk perkembangan teknologi informasi. Harus diperhatikan dalam hal pemanfaatan dan pengamanan yang tepat, agar hasil yang dicapai dapat sesuai dengan tujuan yang memenuhi kebutuhan pengguna teknologi itu sendiri [2]. Metode pengembangan diperlukan sebelum merancang suatu sistem karena dapat memberikan panduan pada setiap langkahnya [3]. Salah satu metode pengembangan sistem yang cukup banyak digunakan adalah *waterfall* yang diterapkan oleh peneliti dalam mengembangkan sistem informasi akademik pondok pesantren.

Pondok pesantren Yapi Sunni merupakan salah satu yayasan dakwah dan pendidikan islam yang terletak di desa Tegalgondo, Wonosari, Klaten. Yapi aktif dalam pengelolaan pendidikan keagamaan dengan visi mencetak generasi Islam yang diharapkan mampu menjadi penerus generasi masa depan yang tangguh dan berilmu. Yapi menaungi dua lembaga pendidikan, yaitu tingkat Tsanawiyah (SMP) dan Aliyah (SMA), hingga saat ini tercatat lebih dari 120 santri dari berbagai daerah untuk menuntut ilmu di pondok pesantren Yapi, seluruh santri diharuskan tinggal di asrama hingga pendidikan yang ditempuh selesai. Selain mengajarkan ilmu keagamaan, pondok ini memberikan ilmu pengetahuan umum sesuai dengan tingkatnya, diharapkan para santri mampu untuk menguasai kedua bidang tersebut sehingga dapat diterapkan dalam kehidupan bermasyarakat dan menjadi bekal untuk jenjang selanjutnya.

Sistem pengelolaan data akademik santri hingga pembuatan laporan, pondok pesantren Yapi menggunakan perpaduan antara pencatatan data secara manual dengan media kertas, arsip, dan *Microsoft Office* yang dalam penerapannya sering muncul kendala saat melakukan pendataan secara manual serta kemungkinan untuk rusak, hilang atau terkena serangan virus cukup tinggi, sehingga memerlukan waktu relatif lama untuk memulihkan kembali data yang terkena virus atau menemukan dokumen hilang. Laporan akademik yang sudah diolah selanjutnya dikirimkan kepada wali santri melalui aplikasi *Whatsapp* atau *SMS* pada akhir semester.

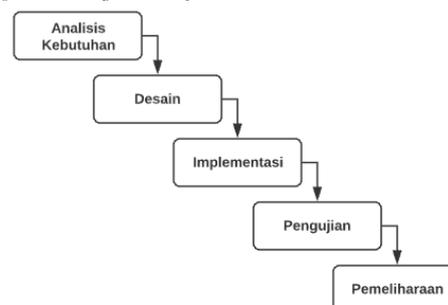
Jumlah santri yang tidak sedikit sering membuat ustadz/pengajar kewalahan dalam mengelola data akademik santri mulai dari nilai, hafalan, hingga pelanggaran dan membutuhkan waktu untuk menyatukannya dalam bentuk laporan sehingga dapat diberikan kepada wali santri. Berdasarkan permasalahan tersebut, dibutuhkan suatu sarana yang mampu melakukan pendataan secara terpadu.

Sejumlah penelitian sejenis mengenai sistem informasi akademik khususnya di pondok pesantren untuk mempermudah, mempercepat kinerja, serta meminimalisir kesalahan yang sering terjadi pada saat penginputan data [4], bertujuan untuk memudahkan staff dalam pengolahan data skor pada setiap bulan dan nilai siswa pada setiap semester [5], pembuatan sistem informasi untuk mengolah informasi dan mengelola data nilai sampai dengan pencetakan raport siswa [6]. Penelitian lainnya mengenai sistem informasi monitoring, bertujuan untuk membantu dan memudahkan pihak sekolah dan wali murid dalam memonitoring kegiatan akademik, akan tetapi bukan diterapkan pada pondok pesantren [7].

Penelitian ini bertujuan untuk rancang bangun sistem informasi akademik pondok pesantren berbasis web yang dapat memberi akses bagi pengelola data akademik santri tingkat aliyah dan menyediakan akses untuk wali santri dalam memantau perkembangan anak mereka. Sistem terdiri dari 3 pengguna yaitu admin, pengajar/ustadz, dan wali santri. Masing-masing pengguna memiliki hak akses yang berbeda. Fasilitas yang disediakan oleh sistem diantaranya adalah *login* dan *CRUD* (*create, read, update, delete*) data santri, jadwal, jadwal kelas, mata pelajaran, hafalan, tahun ajar, user, pelanggaran, tagihan, kelas, dan data ustadz/pengajar. Sistem dapat melakukan pencarian data dan mencetak data ke dalam bentuk pdf. Sistem diancang menggunakan bahasa pemrograman *PHP*, *MySQL* sebagai manajemen databasenya, dan framework *CodeIgniter*. Kemudian dipublikasikan melalui web hosting agar dapat diakses oleh seluruh pengguna. Setelah sistem diterapkan, diharapkan dapat membantu pihak pondok dalam mengelola data akademik dan membantu wali santri untuk mengetahui perkembangan santri.

2. METODE PENELITIAN

Metode analisis deskriptif digunakan untuk menganalisis kebutuhan yang terdapat pada sistem yang akan dibangun [8]. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *waterfall*. Metode ini terdiri dari 5 tahapan yaitu analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan. Tahapan metode *waterfall* ditunjukkan pada Gambar 1.



Gambar 1. Tahapan metode waterfall

2.1 Analisis Kebutuhan

Tahap yang menentukan kebutuhan dari sistem yang dirancang, meliputi kebutuhan fungsional dan kebutuhan non-fungsional sistem. Proses penentuan kebutuhan dilakukan melalui wawancara dengan kepala sekolah tingkat aliyah pondok pesantren Yapi. Hal ini bertujuan agar sistem yang dikembangkan dapat sesuai dengan kebutuhan.

2.1.1 Kebutuhan Fungsional

- Pengguna sistem terdiri dari admin, ustadz/pengajar, dan wali santri yang masing-masing pengguna memiliki hak akses yang berbeda
- Sistem mampu menyediakan fasilitas untuk melakukan *CRUD* (*create, read, update, delete*) dan mencari data yang mencakup data pribadi santri, jadwal, jadwal kelas, mata pelajaran, hafalan, tahun ajar, user, pelanggaran, tagihan, kelas, dan data ustadz/pengajar.
- Sistem mampu menyediakan fasilitas untuk mencetak nilai santri
- Admin dapat mengakses seluruh fasilitas yang disediakan oleh system
- Wali santri dapat melihat nilai, tagihan, pelanggaran, dan hafalan santri

2.1.2 Kebutuhan Non-fungsional

a) Kebutuhan *Software*

Perangkat lunak yang digunakan untuk mengembangkan sistem informasi akademik yaitu menggunakan sistem operasi *Windows/Linux, XAMPP, Visual Studio Code* sebagai media coding, dan *Web Browser* untuk mengakses halaman web.

b) Kebutuhan *Hardware*

Spesifikasi minimal perangkat yang dibutuhkan untuk membangun sistem yaitu komputer/laptop dengan *Processor Intel Core i3*, *RAM 4 GB*, dan *Harddisk 250 GB*. Sedangkan perangkat yang dibutuhkan untuk mengakses sistem informasi akademik pondok pesantren Yapi berbasis web adalah *smartphone* dan *laptop* atau komputer melalui *web browser* yang terhubung dengan jaringan internet.

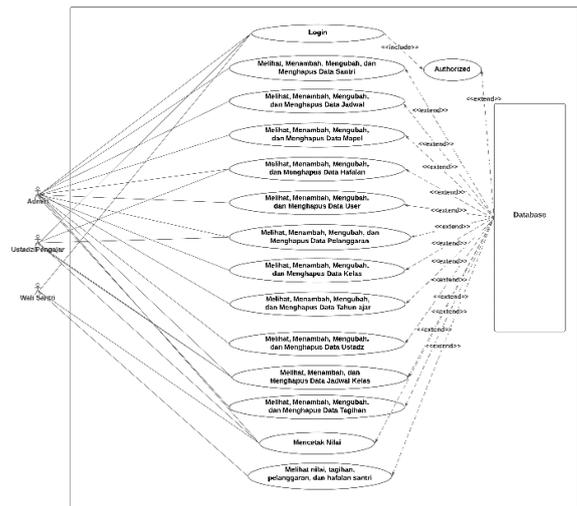
2.2 Desain

Tahap yang menyajikan desain-desain dari aplikasi seperti desain antar muka dan desain *database* diterapkan ke dalam sistem informasi yang dikembangkan [9].

2.2.1 *Usecase Diagram*

Usecase adalah sebuah interaksi sederhana yang menjelaskan tentang identifikasi siapa saja yang dapat menggunakan fungsi dalam sistem tersebut dan urutan yang dilakukan oleh sistem tersebut [10]. *Usecase diagram* menjelaskan manfaat dari sistem jika dilihat dari pandangan

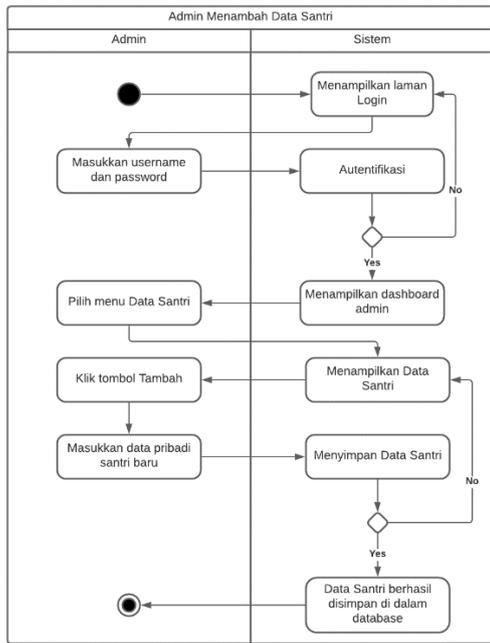
orang yang berada diluar sistem (*actor*) [11]. Pengguna pada sistem informasi akademik pondok pesantren memiliki 3 aktor yang terdiri dari admin, pengajar/ustadz, dan wali santi. Masing-masing aktor memiliki hak akses yang berbeda. Admin memiliki hak akses tertinggi dalam sistem yang dapat melakukan *login, CRUD* data santri, jadwal, jadwal kelas, mata pelajaran, hafalan, tahun ajar, user, pelanggaran, tagihan, kelas, dan data ustadz/pengajar. Pengajar/ustadz memiliki hak akses yang meliputi *login, CRUD* data hafalan, pelanggaran, jadwal kelas, dan cetak nilai. Sedangkan wali santri hanya dapat *login*, melihat tagihan, pelanggaran, hafalan santri, dan mencetak nilai dalam bentuk *pdf*. Interaksi antara pengguna dengan sistem diilustrasikan pada Gambar 2.



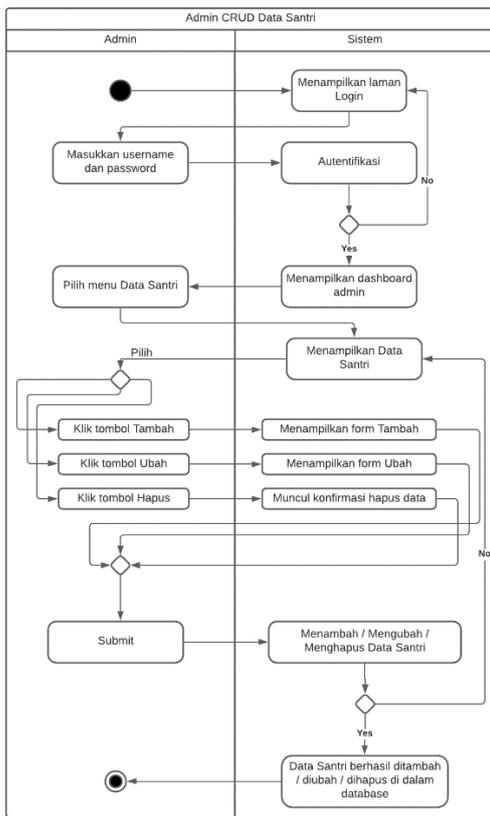
Gambar 2. *Usecase Diagram* Sistem Informasi Akademik pondok pesantren Yapi Sunni Tegalgondo

2.2.2 *Activity Diagram*

Diagram aktifitas menjelaskan alur kerja sistem dalam proses pengolahan data [12]. Admin memiliki hak akses tertinggi dalam mengelola data pada sistem. Sebelum mengelola data, seluruh aktor diwajibkan untuk melakukan *login* terlebih dahulu. Salah satu contoh seperti saat admin ingin menambahkan data santri, diharuskan untuk login ke dalam sistem yang diilustrasikan pada Gambar 3. Sedangkan Diagram aktivitas untuk admin melakukan *CRUD* (*create, read, update, delete*) data santri dapat dilihat pada Gambar 4.



Gambar 3. Activity Diagram login dan insert data santri baru

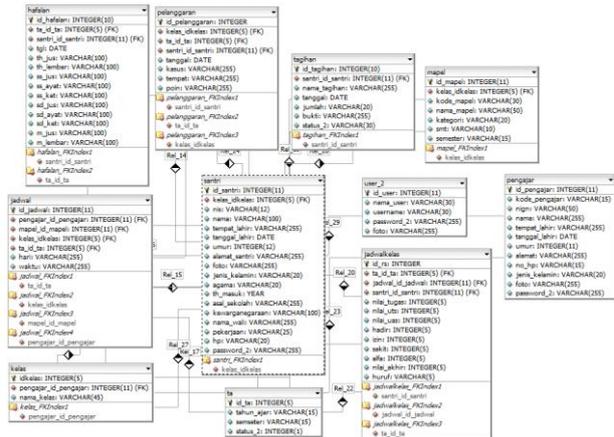


Gambar 4. Activity Diagram login dan CRUD data santri

2.2.3 Rancangan Database

Rancangan database berisi struktur data serta relasi atau hubungan dari setiap data, serta untuk mempermudah pengerjaan basis datanya [5]. Tabel pada

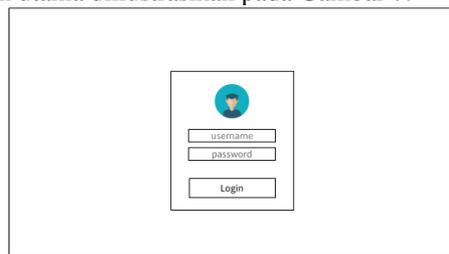
database sistem informasi akademik pondok pesantren ini terdiri dari tabel santri, hafalan, jadwal, kelas, pelanggaran, tagihan, ta, mapel, user, pengajar, dan jadwal kelas. Rancangan basis data dan hubungan antar tabel dalam sistem informasi akademik pondok pesantren Yapi Tegalondo dapat dilihat pada Gambar 5.



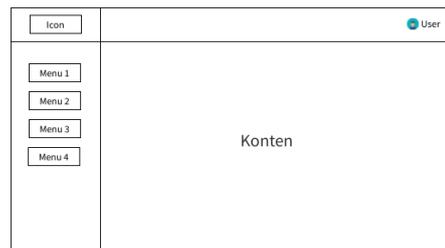
Gambar 5. Rancangan database Sistem Informasi Akademik pondok pesantren Yapi Sunni Tegalondo

2.2.4 Wireframe

Rancangan antarmuka pada sistem informasi akademik pondok pesantren ini terdiri dari halaman login dan halaman utama. Halaman login merupakan tampilan awal saat web diakses yang ditunjukkan pada Gambar 6. Halaman utama terdiri dari sidebar yang berisi menu-menu website, sedangkan konten berisi data yang ditampilkan saat pengguna mengakses menu yang disediakan. Halaman utama diilustrasikan pada Gambar 7.



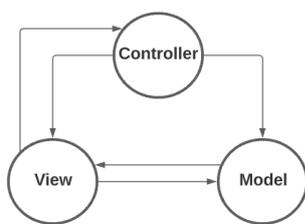
Gambar 6. Rancangan wireframe login



Gambar 7. Rancangan wireframe halaman utama

2.3 Implementasi

Tahap sistem dikembangkan berdasarkan analisis kebutuhan dan desain yang telah ditentukan sebelumnya. Proses implementasi sistem informasi akademik pondok pesantren, peneliti menggunakan *framework CodeIgniter* dan *MySQL* untuk manajemen *databasenya*. *Codeigniter* dibangun menggunakan konsep *Model-View-Controller (MVC) development pattern*. Dalam *Codeigniter*, *browser* berinteraksi melalui *controller*. *Controller* yang akan menerima dan membalas semua permintaan dari *browser*. Ketika *controller* membutuhkan data, maka *controller* akan meminta ke *Model*. Sedangkan untuk tampilan ke *user* akan ditangani oleh *View*. Jadi otak dari aplikasi ada di *controller*, muka aplikasi ada di *view* dan data berada di *model* [13]. Konsep *MVC* diilustrasikan pada Gambar 8.



Gambar 8. Konsep MVC pada *framework CodeIgniter*

2.4 Pengujian

Tahap menguji fungsi dari sistem yang telah dibangun. Tahap ini berguna untuk mengetahui fungsi-fungsi yang dapat bekerja dengan baik begitu pula sebaliknya, sehingga dapat meminimalisir terjadinya *error* saat sistem digunakan. Pengujian yang digunakan pada sistem informasi akademik pondok pesantren yaitu *Black Box Testing* dan *User Acceptance Test (UAT)*.

2.5 Pemeliharaan

Tahap terakhir yang menerapkan sistem informasi akademik pondok pesantren dan memperbaiki *error* yang muncul. Setelah sistem diterapkan, diharapkan dapat membantu pondok pesantren Yapi Sunni Tegalondo dalam mengelola data akademik santri sekaligus wali santri dapat memantau perkembangan dan hasil akademik santri.

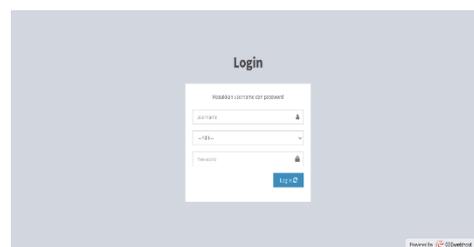
3. HASIL DAN PEMBAHASAN

3.1 Hasil

Sistem informasi akademik ini terdiri dari 3 pengguna yaitu admin, pengajar, dan wali santri. Masing-masing pengguna memiliki hak akses yang berbeda. Sistem menyediakan fasilitas untuk *CRUD (create, read, update, delete)*, *search* dan *print* data.

3.1.1 Halaman Login

Halaman *Login* merupakan halaman awal sebelum masuk ke sistem informasi akademik. Pengguna dapat memasukkan *username*, *password*, dan memilih *role*. Sistem ini memiliki 3 *role* yaitu admin, pengajar, dan santri. Setiap *role* memiliki hak akses yang berbeda. Sistem melakukan verifikasi data pengguna, apabila sesuai diarahkan ke halaman utama masing-masing pengguna. Halaman *Login* ditunjukkan pada Gambar 9.



Gambar 9. Halaman *login* sistem informasi akademik

3.1.2 Halaman Admin

Halaman admin terdapat beberapa menu diantaranya adalah *dashboard* yang menjadi tampilan awal, *profile* yang berisi data pribadi admin, master data untuk mengelola data akademik, manajemen *user* untuk mengelola data admin, *setting* tahun akademik untuk mengatur tahun akademik yang aktif, dan *logout* untuk keluar dari sistem informasi akademik. Master data berisi *menu* data santri, jadwal pelajaran, atur jadwal kelas, mata pelajaran, hafalan, pelanggaran, kelas, tahun ajar, pengajar, dan tagihan. Admin memiliki hak akses tertinggi yang dapat mengakses seluruh fasilitas yang disediakan oleh sistem mulai dari *CRUD (create, read, update, dan delete)* hingga mencetak data ke dalam bentuk *pdf*. Halaman *dashboard* admin ditunjukkan pada Gambar 10.



Gambar 10. Halaman *dashboard* admin

3.1.3 Halaman Pengajar

Halaman pengajar terdapat beberapa menu diantaranya adalah *dashboard* yang merupakan halaman utama pengajar, *profile*, ubah *password*, data akademik, dan *logout*. Menu data akademik

berisi menu data nilai, hafalan, dan pelanggaran. Pengajar dapat melakukan *CRUD* (*create, read, update, dan delete*) nilai berdasarkan mata pelajaran yang diampu, hafalan santri, pelanggaran santri, dan melihat jadwal mengajar berdasarkan tahun akademik yang aktif. Halaman pengajar ditunjukkan pada Gambar 11.



Gambar 11. Halaman jadwal mengajar

3.1.4 Halaman Wali Santri

Halaman wali santri terdapat beberapa menu diantaranya adalah *profile*, ubah *password*, menu melihat informasi akademik yang mencakup nilai, hafalan, pelanggaran, tagihan santri, hasil studi, dan *logout*. Pada halaman tagihan, wali santri dapat melihat status tagihan dan mengunggah bukti pembayaran yang akan dicek

oleh admin untuk diverifikasi. Halaman wali santri dapat dilihat pada Gambar 12.



Gambar 12. Halaman laporan hasil penilaian belajar santri

3.2 Pengujian dan Pembahasan

3.2.1 Pengujian *Blackbox*

Blackbox testing adalah pengujian fungsional *software* tanpa mengetahui struktur internal program [14]. Hasil pengujian menggunakan *blackbox* ditunjukkan pada Tabel 1.

Tabel 1. Hasil pengujian *blackbox*

Pengujian	Bagian Pengujian	Input	Proses	Output	Status
<i>Login</i>					
Mengakses <i>Login</i>	Mengakses halaman <i>login</i>	Memasukkan <i>username, password</i> , dan memilih <i>role</i>	Mengidentifikasi <i>role</i> kemudian memverifikasi <i>username</i> dan <i>password</i> dengan <i>database</i>	Berhasil masuk ke halaman utama Gagal dan kembali ke halaman <i>login</i>	Valid Valid
Admin					
Master Data	Menu Santri	Menampilkan halaman santri kemudian menambah/mengubah/menghapus data	Data santri yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data santri dan menampilkan tabel	Valid
				Gagal menambah/mengubah/menghapus data santri	Valid
	Menu Kelas	Menampilkan halaman kelas kemudian menambah/mengubah/menghapus data	Data kelas yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data kelas dan menampilkan tabel	Valid
				Gagal menambah/mengubah/menghapus data kelas	Valid
	Menu Mapel	Menampilkan halaman mapel kemudian menambah/mengubah/menghapus data	Data mapel yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data mapel dan menampilkan tabel	Valid
				Gagal menambah/mengubah/menghapus data mapel	Valid
	Menu Jadwal	Menampilkan halaman jadwal	Data jadwal yang telah	Berhasil menambah/mengubah/menghapus	Valid

Pelajaran	pelajaran kemudian menambah/mengubah/menghapus data	ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	data jadwal dan menampilkan tabel	Gagal menambah/mengubah/menghapus data jadwal	Valid
Menu Atur Jadwal Kelas	Menampilkan halaman atur jadwal kelas kemudian menambah/menghapus data santri pada jadwal yang dipilih	Data santri yang telah ditambah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/menghapus data dan menampilkan tabel		Valid
			Gagal menambah/mengubah/menghapus data santri pada atur jadwal kelas		Valid
Hafalan	Menampilkan halaman hafalan kemudian menambah/mengubah/menghapus data	Data hafalan yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data dan menampilkan tabel		Valid
			Gagal menambah/mengubah/menghapus data hafalan		Valid
Pelanggaran	Menampilkan halaman pelanggaran kemudian menambah/mengubah/menghapus data	Data pelanggaran yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/menghapus data dan menampilkan tabel		Valid
			Gagal menambah/mengubah/menghapus data pelanggaran		Valid
Tahun Ajar	Menampilkan halaman tahun akademik kemudian menambah/mengubah/menghapus data	Data tahun akademik yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/menghapus data dan menampilkan tabel tahun akademik		Valid
			Gagal menambah/mengubah/menghapus data tahun akademik		Valid
Pengajar	Menampilkan halaman pengajar kemudian menambah/mengubah/menghapus data	Data pengajar yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data dan menampilkan tabel pengajar		Valid
			Gagal menambah/mengubah/menghapus data tahun akademik		Valid
Tagihan	Menampilkan halaman tagihan kemudian menambah/mengubah/menghapus data dan mengubah status tagihan	Data tagihan yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data dan menampilkan tabel tagihan		Valid
			Gagal menambah/mengubah/menghapus data tahun tagihan		Valid
Manajemen User	User	Menampilkan halaman <i>user</i> kemudian menambah/mengubah/menghapus data	Data <i>user</i> yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data dan menampilkan tabel <i>user</i>	Valid
				Gagal menambah/mengubah/menghapus data <i>user</i>	Valid
Tahun Akademik	Tahun Akademik	Menampilkan halaman tahun akademik kemudian mengubah status tahun akademik dengan klik aktifkan	Data tahun akademik telah diaktifkan dan status disimpan dalam <i>database</i>	Berhasil mengubah status tahun akademik	Valid
				Gagal mengubah status tahun akademik	Valid
Pengajar					
Profile	Profil Pengajar	Pilih menu <i>profile</i>	Menampilkan halaman identitas pengajar	Berhasil menampilkan profil pengajar	Valid
				Gagal menampilkan profil pengajar	Valid
Ubah Password	Ubah Password Pengajar	Pilih menu ubah <i>password</i>	Menampilkan <i>form</i> dan memasukkan <i>password</i> baru, lalu simpan	Berhasil mengubah <i>password</i> pengajar	Valid
				Gagal mengubah <i>password</i>	Valid
Data	Nilai	Menampilkan nilai	Data nilai yang telah	Berhasil menambah/mengubah	Valid

Akademik		santri kemudian menambah/mengubah data	ditambah/diubah kemudian disimpan dalam <i>database</i>	data dan menampilkan tabel nilai	
				Gagal menambah/mengubah data nilai	Valid
	Hafalan	Menampilkan halaman hafalan kemudian menambah/mengubah/menghapus data	Data hafalan yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data dan menampilkan tabel	Valid
				Gagal menambah/mengubah/menghapus data hafalan	Valid
	Pelanggaran	Menampilkan halaman pelanggaran kemudian menambah/mengubah/menghapus data	Data pelanggaran yang telah ditambah/diubah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/mengubah/menghapus data dan menampilkan tabel	Valid
				Gagal menambah/mengubah/menghapus data pelanggaran	Valid
Jadwal Mengajar	Jadwal Mengajar	Menampilkan halaman jadwal mengajar kemudian menambah/menghapus data santri pada jadwal yang dipilih	Data santri yang telah ditambah/dihapus kemudian disimpan dalam <i>database</i>	Berhasil menambah/menghapus data dan menampilkan tabel	Valid
				Gagal menambah/mengubah/menghapus data santri pada jadwal mengajar	Valid
Santri/Wali Santri					
Profile	Profil Santri	Pilih menu <i>profile</i>	Menampilkan halaman identitas santri	Berhasil menampilkan profil santri	Valid
				Gagal menampilkan profil santri	Valid
Ubah Password	Ubah Password Santri	Pilih menu ubah <i>password</i>	Menampilkan <i>form</i> dan memasukkan <i>password</i> baru, lalu simpan	Berhasil mengubah <i>password</i> santri	Valid
				Gagal mengubah <i>password</i>	Valid
Nilai	Data Nilai Santri	Pilih menu nilai	Menampilkan data nilai santri berdasarkan semester aktif	Berhasil menampilkan nilai santri	Valid
				Gagal menampilkan nilai santri	Valid
Hafalan	Data Hafalan Santri	Pilih menu hafalan	Menampilkan data hafalan santri	Berhasil menampilkan hafalan santri	Valid
				Gagal menampilkan hafalan santri	Valid
Pelanggaran	Data Pelanggaran Santri	Pilih menu pelanggaran	Menampilkan data pelanggaran santri	Berhasil menampilkan pelanggaran santri	Valid
				Gagal menampilkan pelanggaran santri	Valid
Tagihan	Data Tagihan Santri	Pilih menu tagihan	Menampilkan data tagihan santri dan <i>upload</i> bukti transaksi	Berhasil <i>upload</i> dan menampilkan tagihan santri	Valid
				Gagal <i>upload</i> bukti transaksi pada tagihan santri	Valid
Hasil Studi	Data Hasil Studi Santri	Pilih menu hasil studi, pilih tahun ajar, dan klik simpan	Menampilkan data nilai santri berdasarkan tahun ajar yang dipilih	Berhasil menampilkan nilai santri	Valid
				Gagal menampilkan nilai santri	Valid
Logout					
Logout	Mengakses menu <i>logout</i>	Klik menu <i>logout</i>	Menghapus <i>session</i> dan keluar dari sistem	Berhasil logout dari sistem	Valid
				Gagal logout dari sistem	Valid

3.2.2 Pengujian *User Acceptance Test*

UAT (User Acceptance Test) adalah proses pengujian yang dilakukan oleh pengguna dengan hasil sebuah dokumen hasil uji yang dapat dijadikan bukti dan sudah memenuhi kebutuhan pengguna. Pengguna menguji aplikasi untuk memastikan aplikasi dapat berjalan sesuai fungsinya dan

spesifikasi [15]. Pengujian ini dilakukan dengan memberikan kuesioner berupa 10 pertanyaan kepada 30 responden yang terdiri dari admin, pengajar/ustadz, dan wali santri untuk mengetahui kelayakan sistem. Penilaian dibagi menjadi 5 kategori, yaitu Sangat Setuju (SS) skor = 5, Setuju (S) skor = 4, Netral (N) skor = 3, Tidak Setuju (TS) skor = 2, Sangat Tidak Setuju (STS) skor = 1.

26	4	4	4	5	4	4	5	4	4	4	42
27	5	5	5	5	5	5	5	5	5	5	50
28	5	5	5	5	5	5	5	5	5	5	50
29	5	5	5	5	5	5	5	5	5	5	50
30	5	5	5	5	5	5	5	5	5	5	50
rx_y	0,83	0,88	0,91	0,93	0,98	0,98	0,90	0,84	0,94	0,94	
	8929	3911	5473	1534	1957	1957	0432	6861	2733	2733	
	5924	2222	1802	1131	3883	3883	9273	9598	0382	0382	
t hitung	8,15	10,0	12,0	13,5	27,4	27,4	10,9	8,42	14,9	14,9	
	6692	0143	3897	5468	7734	7734	5333	6221	5578	5578	
	684	524	018	623	707	707	173	677	14	14	
t tabel	1,70										
(95%, 28)	1130										
	934										
keterangan	valid										
n											
Jml valid	10										

Hasil uji validitas menggunakan korelasi *Pearson* dan Uji *t* menunjukkan bahwa 10 pertanyaan yang diberikan kepada 30 responden valid. Hasil jawaban dari kuesioner dihitung menggunakan rumus 3 [19].

$$P = \frac{\text{Total}}{n\text{Max}} \times 100\% \quad (3)$$

Keterangan :

P = Presentase

Total = Jumlah jawaban x skor tiap jawaban

nMax = Jumlah responden x skor tertinggi

Responden yang mengisi kuesioner berjumlah 30 dan skor tertinggi adalah 5, maka diperoleh nMax = 30 x 5 = 150. Hasil perhitungan berdasarkan jawaban dari pengguna dapat dilihat pada Tabel 4.

Tabel 4. Hasil Perhitungan Jawaban Kuesioner

Kode	Nilai					Total	%
	SSx5	Sx4	Nx3	TSx2	STSx1		
Q1	110	32	0	0	0	142	94%
Q2	105	36	0	0	0	141	94%
Q3	100	40	0	0	0	140	93%
Q4	100	40	0	0	0	140	93%
Q5	95	44	0	0	0	139	92%
Q6	95	44	0	0	0	139	92%
Q7	105	36	0	0	0	141	94%
Q8	90	44	3	0	0	137	91%
Q9	90	48	0	0	0	138	92%
Q10	90	48	0	0	0	138	92%
Rata-rata presentase							93%

Berdasarkan hasil pengujian UAT yang dilakukan oleh pengguna, diperoleh rata-rata presentase sebesar 93% yang dapat disimpulkan

bahwa sistem informasi akademik dapat diterima dan digunakan di pondok pesantren Yapi Sunni.

4. PENUTUP

Berdasarkan hasil pengujian *blackbox*, sistem informasi akademik pondok pesantren Yapi Sunni dapat berjalan dengan baik. Sistem ini terdiri dari 3 pengguna yaitu admin, pengajar, dan wali santri. Sistem dapat digunakan untuk mengelola data akademik santri tingkat aliyah dan memberi akses bagi wali santri untuk memantau perkembangan akademik santri. Hasil pengujian *UAT* yang berasal dari 30 responden yang terdiri dari admin, pengajar, dan wali santri diperoleh rata-rata presentase sebesar 93% , maka dapat disimpulkan sistem sesuai dengan kebutuhan pengguna yang dapat memberikan kemudahan bagi pengelola data akademik santri, memberi akses bagi wali santri untuk memantau perkembangan santri, dan dapat meminimalisir kesalahan saat menginputkan data. Saran dari peneliti untuk penelitian berikutnya, system dapat dikembangkan lebih lanjut dengan menambahkan fitur notifikasi pada tagihan.

DAFTAR PUSTAKA

- [1] B. rahma. Ary Dwi Pantoro, Ucu Darusalam, "Academic Information System for Student Leave at the National University," *J. Mantik*, vol. 3, no. January, pp. 31–38, 2019.
- [2] M. Agnes, L. Jola, and S. Gaspersz, "Academic Information System for Student (Case Study: Victory University of Sorong)," *Int. J. Comput. Appl.*, vol. 180, no. 43, pp. 26–33, 2018, doi: 10.5120/ijca2018917134.
- [3] T. K. Rahayu, Susanto, and Suwarjono, "Application Report Process of Islamic School Based on Pesantren Boarding Using Waterfall Model," *J. Phys. Conf. Ser.*, vol. 1569, no. 2, 2020, doi: 10.1088/1742-6596/1569/2/022025.

- [4] A. Fajar, R. T. Prasetyo, A. R. Sanjaya, and A. R. Sanjaya, "SISTEM INFORMASI AKADEMIK PESANTREN (SIAKSEN) BERBASIS WEB PADA PESANTREN," vol. 2, no. 1, pp. 92–101, 2021.
- [5] A. Lutfi, "Sistem Informasi Akademik Madrasah Aliyah Salafiyah Syafi'iyah Menggunakan Php Dan Mysql," *AiTech*, vol. 3, no. 2, pp. 104–112, 2017.
- [6] A. Supriyatna and D. Puspitasari, "Implementation of Extreme Programming Method in Web Based Digital Report Value Information System Design," *IJISTECH (International J. Inf. Syst. Technol.)*, vol. 5, no. 1, p. 67, 2021, doi: 10.30645/ijistech.v5i1.116.
- [7] E. Sutinah, G. N. Azima, and E. F. Imaduddin, "Sistem Informasi Monitoring Akademik Dan Prestasi Siswa Dengan Metode Waterfall," *J. Inf. Eng. Educ. Technol.*, vol. 2, no. 1, p. 47, 2018, doi: 10.26740/jieet.v2n1.p47-59.
- [8] A. R. Kurniawan and L. Warlina, "Web Service for Academic Information Systems," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 879, no. 1, 2020, doi: 10.1088/1757-899X/879/1/012009.
- [9] Y. Firmansyah and U. Udi, "Penerapan Metode SDLC Waterfall Dalam Pembuatan Sistem Informasi Akademik Berbasis Web Studi Kasus Pondok Pesantren Al-Habib Sholeh Kabupaten Kubu Raya, Kalimantan Barat," *J. Teknol. dan Manaj. Inform.*, vol. 4, no. 1, 2017, doi: 10.26905/jtmi.v4i1.1605.
- [10] A. T. Setyowinarti and Y. I. Kurniawan, "Sistem Penjadwalan Shift Jaga di PT . Air Mancur Berbasis Web dan SMS Gateway," vol. 19, no. 01, pp. 16–21, 2019.
- [11] A. Burrahman, "Membangun Sistem Informasi Akademik Berbasis Web Pada Pondok Pesantren Salafiyah Al-Baqiyatussa'Diyyah Tembilahan," *Sistemasi*, vol. 6, no. 1, p. 33, 2018, doi: 10.32520/stmsi.v6i1.26.
- [12] A. Zuliyana and D. A. Anggoro, "Sistem Informasi Akademik Sekolah berbasis Web di SMK Widya Taruna Kabupaten Karanganyar," *Emit. J. Tek. Elektro*, vol. 20, no. 02, pp. 96–103, 2020, doi: 10.23917/emitor.v20i02.10942.
- [13] A. Subari, D. Y. Tadeus, H. Winarno, and T. Yuwono, "Rancang Bangun Sistem Administrasi Kerja Praktek Dan Tugas Akhir Berbasis Web Menggunakan Framework Codeigniter," *Gema Teknol.*, vol. 19, no. 4, p. 1, 2018, doi: 10.14710/gt.v19i4.19147.
- [14] D. I. Permatasari, "Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 1, p. 135, 2020, doi: 10.26418/justin.v8i1.34452.
- [15] B. Harijanto, I. K. Putri, and H. N. Javier, "Penerapan Metode User Centered Design (UCD) Katalog Online Berbasis Web di Rumah Makan Alfa Lesehan Kabupaten Blitar," *Semin. Inform. Apl. Polinema – 2021*, pp. 147–156, 2021.
- [16] N. A. Murti, "Analisis Usability Testing Pada Aplikasi Transportasi Online Untuk Mengukur Kepuasan Pengguna," *JSil (Jurnal Sist. Informasi)*, vol. 7, no. 1, p. 19, 2020, doi: 10.30656/jsii.v7i1.2102.
- [17] T. Cahyono, "Statistik Uji Korelasi," *J. Chem. Inf. Model.*, vol. Pertama, pp. 1–77, 2017.
- [18] Sugiyono, *Statistika Untuk Penelitian*. Bandung: Penerbit ALFABETA, 2014.
- [19] T. Widyasari, "Sistem Informasi Penjualan Sepeda Motor Di Dealer Sinar Mas Makmur Motor," vol. 21, no. 01, pp. 1–9, 2018.