

BAB II

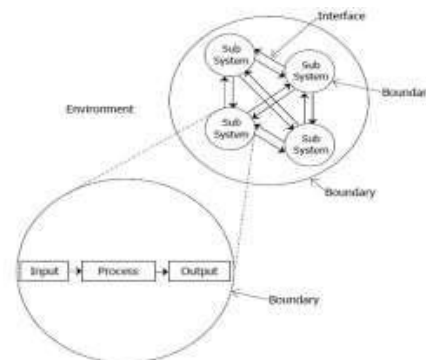
TINJAUAN PUSTAKA

2.1 Sistem

Suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variable-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. (Hidayat, A.& Piliang, F. (2019).

. Suatu kesatuan yang terdiri dari komponen dan elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. (Febriani, O. M., Nugroho, H. W., Firdhayanti, A., & Rahardi, A. 2022, August).

Suatu sistem juga mempunyai sebuah karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan tujuan (*goal*).

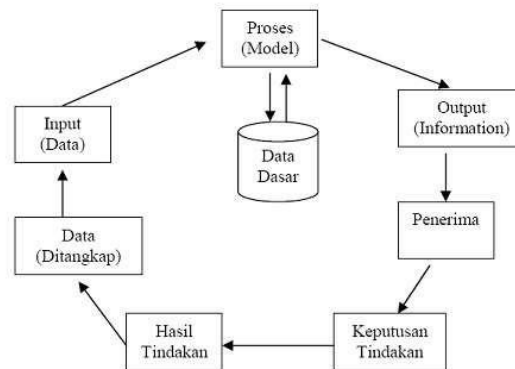


Gambar 2.1 Karakteristik Sistem

2.2 Informasi

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan kejadian-kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan.

Informasi adalah data yang telah diolah menjadi sebuah bentuk berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang. (Sarmidi, dan Bhui, B. 2019).



Gambar2.2 SiklusInformasi

2.3 Sistem Informasi

Sistem informasi yang terdiri dari komponen-komponen di atas disebut dengan block), yaitu blok masukan (input block), blok model (model block), blok keluaran (output block), blok teknologi (technology block), dan blok kendali (control block). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sarannya. (Hidayat, A. T, 2019)

2.4 HTML

HyperText Markup Language (HTML) adalah bahas markup yang digunakan untuk membuat situs web, menampilkan berbagai informasi di browser web, dan hanya memformat hypertext ke file ASCII untuk tampilan terintegrasi. Dengan kata lain file dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII biasa sehingga menjadi halaman beranda dengan perintah HTML. Berawal dari bahasa yang sebelumnya banyak digunakan didunia penerbitan dan percetakan yang disebut SGML (*Standard Generalized Markup Language*), HTML merupakan standar yang banyak digunakan untuk menampilkan halaman web. HTML sekarang menjadi standar internet yang ditentukan dan dikontrol oleh penggunaannya *World Wide Web Consortium*(W3C).

2.5 Pemodelan Berorientasi Objek

2.5.1 Pengertian Pemodelan Berorientasi Objek

Pemodelan berorientasi objek merupakan adalah suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi atau sistem lainnya). Sistem berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek – objek dunia nyata. Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahapan analisis, perancangan, pemrograman dan pengujian perangkat lunak. Ada berbagai teknik yang digunakan pada masing–masing tahapan tersebut, dengan aturan dan pemodelan tertentu. Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem dapat mewarisi sifat atribut dan sifat dari komponen lainnya dan dapat berinteraksi satu sama lainnya. Rosa A.S. dan M. Shalahuddin dalam bukunya Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek (Rosa A.S. dan M. Shalahuddin, 2018)

1.5.2 Ciri – Ciri Pemodelan Berorientasi Objek

1. Kelas (*class*)

Kelas adalah kumpulan objek – objek dengan karakteristik yang sama. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi atau metode), hubungan (relationship) dan arti. Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode pemrograman yang merupakan metodologi berorientasi objek. Kelas secara fisik adalah berkas atau file yang berisi kode program, dimana kode program merupakan semua hal yang terkait dengan nama kelas. (Rosa A.S. dan M. Shalahuddin, 2018)

2. Objek (*object*)

Objek adalah suatu yang mewakili dunia nyata seperti benda, manusia, tempat, struktur, status atau hal lainnya yang abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau berpengaruh pada suatu objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

Secara teknis, sebuah kelas saat program dieksekusi maka akan dibuat sebuah objek. Objek dilihat dari segi teknis elemen pada saat runtime yang akan diciptakan, dimanipulasi dan dihancurkan saat dieksekusi maka sebuah objek hanya ada pada saat sebuah program dieksekusi. (Rosa A.S. dan M. Shalahuddin, 2018).

3. Metode (*method*)

Operasi atau metode merupakan fungsi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode berasal dari event, aktivitas atau aksi keadaan, fungsi atau kelakuan dunia nyata. Contoh metode misalnya read, write, move, copy dan sebagainya. Kelas sebaiknya memiliki metode get dan set untuk setiap atribut agar konsep enkapsulasi tetap terjaga. Metode get digunakan untuk memberi akses kelas lain dalam mengakses atribut, dan set adalah metode yang digunakan untuk mengisi atribut, agar kelas lain tidak mengakses atribut secara langsung. (Rosa A.S. dan M. Shalahuddin, 2018)

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen – elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dimiliki secara individu oleh sebuah objek. Atribut sebaiknya bersifat private untuk menjaga konsep enkapsulasi. (Rosa A.S. dan M. Shalahuddin, 2018)

5. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan pelayanan (operasi) yang dimiliki oleh objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya. (Rosa A.S. dan M. Shalahuddin, 2018).

6. Antarmuka (*interface*)

Antarmuka atau interface sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain. Metode pada antar muka yang diimplementasikan pada suatu kelas harus sama persis dengan yang ada pada antarmuka. Antarmuka atau interface biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas tapi mengakses antarmukanya. (Rosa A.S. dan M. Shalahuddin, 2018)

7. Package

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas – kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda. (Rosa A.S. dan M. Shalahuddin, 2018)




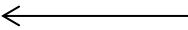
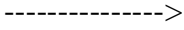
2.5.3 UML (*Unified Modeling Language*)

UML adalah: *Unified Modeling Language (UML)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. (Rosa & Shalahuddin, 2018)

2.5.3.1 *Use Case Diagram*

Merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. (Rosa A.S dan M. Shalahuddin, 2018)

Tabel 2.1 Simbol *Usecase Diagram* (Rosa A.S dan M. Shalahuddin, 2018)

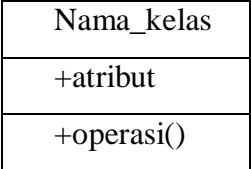



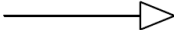
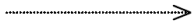

No	Simbol	Keterangan
1.	<p><i>UseCase</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2.	<p><i>Actor</i></p> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, walaupun simbol aktor adalah orang namun aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<p><i>Association</i></p> 	Komunikasi antara aktor dan use case yang berpartisipasi pada usecase atau use case memiliki interaksi dengan aktor.
4.	<p>Ekstensi <i>Extend</i></p>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu.
5.	<p><i>Generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case, fungsi yang satunya fungsi yang lebih umum dari lainnya.
6.	<p><i>Include</i></p> 	Relasi use case tambahan ke sebuah use case dimana use case ditambahkan memerlukan use case ini untuk menjalankan fungsinya.

2.5.3.2 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. (Rosa A.S dan M. Shalahuddin, 2018).

Sebuah kelas diagram terdiri dari sejumlah kelas yang dihubungkan dengan garis yang menunjukkan hubungan antar kelas.

Tabel 2.2 Simbol *Class Diagram*(Rosa A.S dan M. Shalahuddin, 2018)


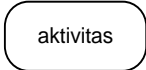



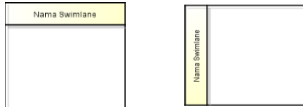
Simbol	Keterangan
<p>KelasClass</p> 	Kelas pada struktur system
<p>Antar mukainterface</p> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek
<p>Asosiasiassociation</p> 	Relasi antar kelas dengan makna umum ,asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarahdirected</p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
<p>Kebergantungan atau Dependency</p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
<p>Agregasiaggregation</p> 	Relasi antar kelas dengan makna semua-bagian(whole-part)

2.5.3.3 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa A.S dan M. Shalahuddin, 2018)

Berikut ini adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol *Activity Diagram*(Rosa A.S dan M. Shalahuddin, 2018)



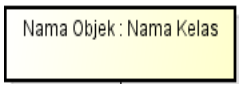

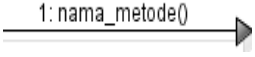
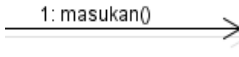
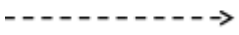
Simbol	Keterangan
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.5.3.4 *Sequence Diagram*

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek atau message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki kelas. (Rosa A.S dan M. Shalahuddin, 2018)

Berikut ini adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.4 Simbol *Sequence Diagram* (Rosa A.S dan M. Shalahuddin, 2018)

Simbol	Keterangan
<p>Aktor</p>  <p>atau</p> <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.</p>
<p>Garis hidup</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatukan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>
<p>Pesan tipe <i>create</i> <code><<create>></code></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i> <code>1: nama_metode()</code></p> 	<p>Menyatakan suatu objek memanggil operasi metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe <i>send</i> <code>1: masukan()</code></p> 	<p>Menyatakan bahwa suatu objek mengirim data masukan informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <code>1: keluaran</code></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan sesuatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.</p>
<p>Pesan Tipe <i>Destroy</i></p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

2.6 Alat Implementasi Sistem

2.6.1 MySQL

(Loka Dwirtara) MySQL bersifat RDBMS (Relational Database Management Sistem) yang memungkinkan seorang admin dapat menyimpan banyak informasi ke tabel-tabel, dimana tabel-tabel tersebut saling berkaitan satu sama lain. Keuntungan RDBMS sendiri adalah kita dapat memecah database ke dalam tabel-tabel yang berbeda, setiap tabel memiliki informasi yang berkaitan dengan tabel yang lainnya.

MySQL adalah *multi user database* yang menggunakan bahasa *Structured Query Language* (SQL). MySQL dalam operasi *client-server* melibatkan server daemon MySQL di sisi *server* dan sebagai macam program serta library yang berjalan di sisi client MySQL mampu menangani data yang cukup besar.

1.6.2 PHP

PHP (*Hypertext Preprocessor*) adalah pemrograman untuk dijalankan melalui halaman web, umumnya digunakan untuk mengolah informasi di internet. Sedangkan dalam pengertian lain PHP adalah singkatan dari Hypertext Preprocessor yaitu bahasa pemrograman *webserverside* yang bersifat *open source* atau gratis. PHP merupakan script yang menyatu dengan HTML dan berada pada server. (Pahlevi, O., Mulyani, A., & Khoir, M. 2018).

1.6.3 Xampp

Fungsi Xampp sendiri adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: *Apache HTTP Server*, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama Xampp sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan

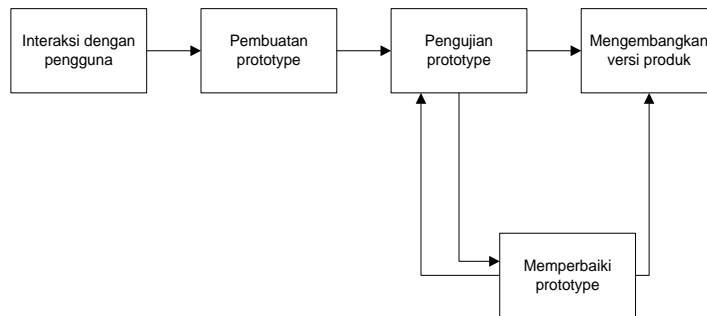
halaman web yang dinamis. Untuk mendapatkannya Xampp anda dapat mendownload langsung dari web resminya.

2.7 Metode Pengembangan Sistem

Metodelogi pengembangan sistem pada penelitian ini menggunakan metode *Prototype*. Menurut (Adi Nugroho, 2018) metode *prototype* dibuat saat pengguna tidak tahu pasti apa yang mereka inginkan baik rincian masukannya, rincian proses dan rincian keluaran yang diinginkan untuk itu dibuatlah *Prototype* kepada pengguna. Kemudian pengguna menyarankan perbaikan-perbaikan jika terdapat kekurangan sistem yang perlu diperbaiki. (Karim, A. S., & Rahardi, A. 2021)

Adapun tahapan-tahapannya adalah sebagai berikut :

1. **Interaksi dengan pengguna**
Pada tahapan ini penyusun menganalisis apa yang ingin pengguna dapatkan dari sistem/perangkat lunak itu. Sehingga aplikasi yang dihasilkan sesuai dengan kebutuhan pengguna dan sistem.
2. **Membuat *Prototype***
Pada tahapan ini akan dibuat sebuah *Prototype* aplikasi berbasis web berdasarkan atas kebutuhan pengguna dan sistem pada tahap interaksi dengan pengguna.
3. **Menguji *Prototype***
Tahapan ini adalah proses penilaian terhadap *Prototype* yang telah dibuat apakah sesuai dengan kebutuhan atau tidak jika tidak maka *Prototype* akan diperbaiki.
4. **Memperbaiki *Prototype***
Setelah ditemukan letak kesalahan dari *Prototype* yang dirancang pada tahapan ini penyusun akan membuat atau memperbaiki *Prototype* yang ada setelah itu akan di uji lagi sehingga *Prototype* sesuai dengan keinginan pengguna.
5. **Mengembangkan Versi Produk** setelah aplikasi dapat berjalan dan memenuhi kebutuhan sistem maka aplikasi ini siap dip



Gambar 2.3 Metode *Prototype*

Prototype memiliki tahapan tahapan sebagai berikut :

a. Identifikasi kebutuhan *end* Penduduk

Pada tahap ini para pemakai akhir mengidentifikasi kebutuhan bisnis mereka dan menilai kelayakan beberapa alternatif solusi sistem informasi. Pengguna sistem informasi dan vendor mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat beserta alternatif solusi sistem.

b. Membangun *prototype business system*

Pada tahap ini para pemakai akhir atau pakar sistem informasi menggunakan alat pengembangan aplikasi untuk secara interaktif mendesain dan menguji *prototype* berbagai komponen sistem informasi yang memenuhi kebutuhan para pemakai akhir. Membangun *prototyping* aplikasi pengembangan dengan membuat model sebagai uji coba yang mewakili kebutuhan pengguna secara garis besar.

c. Revisi *prototype* kedalam bentuk yang mendekati kebutuhan *end* Penduduk

Model sistem bisnis diuji coba, dievaluasi dan dimodifikasi berulang-ulang hingga dapat diterima oleh pengguna dan dirasakan oleh pengguna telah sesuai dengan kebutuhan.

d. Menggunakan dan memelihara *business system* yang telah diterima

Dalam tahap ini sistem bisnis yang telah disepakati antara pengguna dan *vendor* dapat dimodifikasi dengan mudah karena sebagian besar dokumentasi dari sistem telah tersimpan. Model *prototype* dimulai dengan pengumpulan kebutuhan.

Pengembang dan Penduduk bertemu dan mendefinisikan obyektif keseluruhan dari *software*, mengidentifikasi segala kebutuhan yang diketahui dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan perencanaan secara cepat. Sehingga prototyping dikenal dengan Rapid Application Development (RAD).

2.8 Tinjauan Studi

Dalam penelitian ini akan menggunakan tiga tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu:

Tabel 2.4 Ringkasan Tinjauan Studi

Nama	Judul	Fitur	Metode Penelitian	Hasil
Irfan Muhammad Abyan	Sistem Informasi Pondok Pesantren Berbasis Web	Aplikasi Sistem Informasi Pondok Pesantren merupakan sebuah sistem informasi yang didesain dan dibuat khusus untuk menyesuaikan kegiatan belajar mengajar pada Pondok Pesantren yang diharapkan dapat membantu proses kegiatan secara keseluruhan.	Prototype	memudahkan dalam pengumpulan dan pengelolaan informasi – informasi yang dihasilkan dari proses belajar mengajar dalam satu tempat yang mudah untuk diakses dan disebarakan kembali

Tabel 2.4 Ringkasan Tinjauan Studi (lanjutan tabel)

<p>Slamet Rahayu Mauliyatul Ameliah</p>	<p>Rancang Bangun Sistem Informasi Pendaftaran Santri Pondok Pesantren</p>	<p>Pendaftaran Santri Baru</p>	<p><i>UML (Unified Modelling Language)</i></p>	<p>sistem informasi pendaftaran yang dapat menampilkan informasi terbaru tentang pesantren yang mudah diakses oleh masyarakat. Penyebaran informasi tersebut tidak hanya pada <i>website</i> tapi juga pada berbagai media sosial seperti <i>twitter</i> dan <i>facebook</i></p>
<p>Ahmad Bagus Setiawan1, Juli Sulaksono2, Resty Wulanningrum3</p>	<p>Penerapan Sistem Informasi Berbasis Website di Pondok Pesantren Kota Kediri</p>	<p>pengurus pondok dapat mengetahui data Santri, rekap pembayaran, kurikulum dan data gedung yang dimiliki oleh pondok pesantren</p>	<p>Prototype</p>	<p>Dengan adanya Sistem Informasi berbasis website di Pondok Pesantren, pengurus pondok dapat mengetahui data Santri, rekap pembayaran, kurikulum dan data gedung yang dimiliki oleh pondok pesantren</p>