

PENGEMBANGAN *WEB SERVICE ONLINE* PEMBAYARAN *ONLINE* MENGGUNAKAN *SIMPLE OBJECT ACCESS PROTOCOL (SOAP)* (STUDY KASUS : SEKOLAH DASAR ISLAM AL-AZHAR BSD)

Nurhayati¹, Hendra Bayu Suseno², Andi Malik Rizki Maulana³

^{1,2,3} Program Studi Teknik Informatika Fakultas Sains dan Teknologi, UIN Syarif Hidayatullah Jakarta

¹nurhayatibuslim@gmail.com

²hendra.bayu@uinjkt.ac.id

³andi.malik.notifications@gmail.com

Abstrak: Sekolah Islam Al-Azhar Bumi Serpong Damai (Al-Azhar BSD) telah menjalin kerjasama dengan bank-bank rekanan dalam menyelenggarakan sistem pembayaran online. Al-Azhar BSD menyediakan web service untuk komunikasi data antara bank-bank rekanan dengan Al-Azhar BSD. Ada tiga kebutuhan yang belum terpenuhi oleh sistem ini. Pertama, Al-Azhar BSD menginginkan pembayar dapat membayar beberapa tagihan cukup dengan satu transaksi. Kedua, bagian keuangan menginginkan adanya pengolahan data transaksi yang masuk menjadi rekapitulasi harian. Ketiga, bagian keuangan dan yayasan juga menginginkan ketersediaan informasi perkembangan pembayaran. Untuk memenuhi kebutuhan-kebutuhan tersebut, perlu dilakukan perbaikan terhadap web service dengan mempertahankan spesifikasi yang telah disepakati bersama oleh Al-Azhar BSD dan bank-bank rekanan. Web service sistem ini dikembangkan menggunakan bahasa pemrograman PHP, database yang digunakan adalah MySQL, dan mengikuti spesifikasi SOAP untuk pertukaran message dengan sistem bank rekanan. Metode yang digunakan untuk pengembangan sistem adalah metode Extreme Programming (XP) dengan jumlah iterasi sebanyak tiga iterasi. Pengembangan web service ini mampu meningkatkan rata-rata transaksi per bulan pada sistem pembayaran online yang tadinya hanya sekitar 33 transaksi per bulan menjadi 2.324 transaksi per bulan. Artinya banyak pembayar yang beralih menggunakan metode pembayaran online.

Kata Kunci: web service, pembayaran online, SOAP, PHP, MySQL

Abstract: *Al-Azhar Bumi Serpong Damai (BSD) Islamic elementary school have been in association with various banks to accommodate online payment. Al-Azhar BSD provided web service for data communication between Al-Azhar partner banks. There were three requirement that this system can not handle, first, various payment with one transaction, second, daily recapitulation for financial department, and the last is payment information for Al-Azhar foundation. In order to meet this requirement the web service need some improvement without changing Al-Azhar and Banks specification. This web service developed using PHP programing language, MYSQL database, ans SOAP specification in order to exchange message between system and banks. Development method using three iteration Extreme Programing Method (XP). This web service development increase transaction number average for each month. From 33 transaction month became 2.324 transaction each month.*

Which means a lot of people moved to used online payment system.

Key Word : Web Service, Online Payment, SOAP, PHP, My SQL

I. PENDAHULUAN

Sistem pembayaran online Al-Azhar Bumi Serpong Damai (Al-Azhar BSD) diselenggarakan atas kerja sama Sekolah Islam Al-Azhar BSD dengan bank rekanan. Al-Azhar BSD menyediakan web service untuk bertukar informasi tagihan dan pembayaran dengan bank. Bank rekanan menyediakan fasilitas pembayaran. Saat ini ada dua bank yang telah bekerja sama yaitu Bank Muamalat Indonesia (BMI) dan Bank Rakyat Indonesia Syariah (BRIS).

Sistem pembayaran *online* sudah berjalan sesuai dengan fungsinya, namun tidak sepenuhnya berjalan sesuai dengan kebutuhan penggunanya. Beberapa kebutuhan yang belum dipenuhi oleh sistem ini adalah:

1. Pembayaran dapat dilakukan dengan nilai berapa pun bagi siswa atau orangtua/wali,
2. Rekapitulasi (rekap) harian pembayaran bagi kasir bagian keuangan, dan
3. Pemantauan perkembangan pembayaran bagi bagian keuangan dan yayasan.

Untuk memenuhi kebutuhan-kebutuhan yang belum dapat dipenuhi sistem pembayaran online yang sudah berjalan, penulis melaksanakan penelitian di Sekolah Islam Al-Azhar BSD dan mengambil judul “PENGEMBANGAN *WEB SERVICE* PEMBAYARAN *ONLINE* MENGGUNAKAN *SIMPLE OBJECT ACCESS PROTOCOL* (SOAP) PADA SEKOLAH ISLAM AL-AZHAR BSD”.

II. LANDASAN TEORI

SOAP adalah protocol pertukaran informasi dalam lingkungan yang terdesentralisasi dan terdistribusi. SOAP adalah protocol berbasis XML yang terdiri dari tiga bagian [1]. Sebuah amplop yang mendefinisikan *framework* untuk mendeskripsikan apa yang ada didalam pesan dan bagaimana memproses pesan tersebut.

1. Sebuah himpunan aturan pengkodean untuk menyatakan instans-instans dari tipe data yang didefinisikan oleh aplikasi, dan
2. Sebuah konvensi untuk menyajikan *remote procedure call* dan *respons*.

SOAP bisa digunakan dengan menggabungkan dengan bermacam-macam protokol, namun pada penelitian ini dibahas penggunaan SOAP bersama-sama dengan HTTP dan HTTP Extension *Framework*.

Pada penelitian ini juga digunakan WSDL WSDL adalah format XML untuk mendeskripsikan jasa jaringan sebagai sebuah himpunan *end point – end point* yang beroperasi pada pesan yang berisikan baik informasi yang *document-oriented* atau informasi yang *procedure-oriented* [2]. Operasi-operasi dan pesan-[esan dideskripsikan secara abstrak, dan kemudian terikat ke sebuah protocol jaringan nyata dan format pesan untuk mendefinisikan sebuah *endpoint*. *Endpoint-endpoint* nyata yang berhubungan digabungkan ke dalam *endpoint-endpoint* abstrak (*services*). WSDL ini dapat diperluas untuk mengizinkan adanya deskripsi endpoint-endpoint dan pesan-pesannya tanpa melihat format pesan atau protocol-protokol jaringan yang digunakan untuk berkomunikasi. WSDL pada penelitian ini adalah dalam hubungannya dengan SOAP.

III. METODOLOGI

3.1. Metode Pengumpulan Data

Penulis mengumpulkan data dan informasi yang dibutuhkan dalam penelitian ini dengan melakukan wawancara dan studi pustaka.

1. Wawancara

Penulis mewawancarai *customer* untuk memperoleh informasi mengenai kebutuhan-kebutuhan sistem yang akan dikembangkan.

2. Studi Pustaka

Penulis mempelajari metode *extreme programming*, proses *test-driven development*, pemodelan UML, dan teknologi *web service* dari buku-buku, berkas-berkas elektronik, dan artikel-artikel di *Internet*.

3.2. Metode Pengembangan Sistem

Metode yang penulis gunakan dalam pengembangan sistem ini adalah metode *extreme programming*. Metode ini digunakan karena

pendekatan XP memungkinkan umpan balik yang cepat dari *customer* serta memungkinkan *customer* untuk mengetahui perkembangan pengembangan sistem pada setiap iterasi. Selain itu, metode ini juga menerima perubahan kebutuhan.

Pengembangan sistem terdiri dari perencanaan, analisis, perancangan, *coding*, pengujian, dan pengiriman. Iterasi terjadi sebanyak tiga kali pada tahap analisis, perancangan, *coding*, dan pengujian. Berikut ini adalah aktivitas-aktivitas yang dilakukan pada setiap tahap.

1. Perencanaan

Penulis bersama *customer* merencanakan story. *Customer* membuat *story*, kemudian penulis membuat perkiraan waktu untuk setiap *story*.

2. Analisis

Penulis bertanya kepada *customer* secara langsung untuk memperoleh informasi mengenai kebutuhan-kebutuhan sistem.

3. Perancangan

Penulis merancang sistem dengan menggunakan UML. Penulis juga membuat struktur tabel-tabel yang diperlukan untuk menyimpan data dalam *database*.

4. *Coding*

Penulis mengimplementasikan sistem ke dalam *coding* dengan mengacu pada kebutuhan dan perancangan yang dibuat.

5. Pengujian

Sistem diuji dengan *pilot testing*.

6. Pengiriman

Penulis mengirim aplikasi yang telah dibangun kepada *customer* melalui surat elektronik.

3.3. Perangkat Pengembangan

3.3.1. Perangkat Keras

Perangkat keras yang digunakan untuk mengembangkan sistem ini adalah komputer *laptop* yang spesifikasinya ada pada Tabel 1.

Tabel 7. Spesifikasi Perangkat Keras

Prosesor	Intel(R) Core(TM)2 Duo CPU, T6600 @2.20 GHz
Chipset	SIS M672 + SIS968 <i>Chipset</i>
Sistem Grafik	SIS Mirage 3 sampai 256MB
Memori	DDRII SDRAM 200-pin SO-DIMM modul sampai 4 GB menggunakan modul 533/667/800 MHz DDRII SDRAM.
Sistem Audio	<i>High Definition Audio</i> , kompatibel dengan <i>Direct Sound 3D</i> , <i>S/PDIF Digital</i> , <i>Built-in Microphone</i> dan dua <i>speaker</i> .
USB	Tiga <i>port</i> USB 2.0
Express Card	1 <i>slot</i>
Wireless LAN	802.11b/g
Inside BIOS	Phoenix BIOS <i>Technology</i> .

3.3.2. Perangkat Lunak

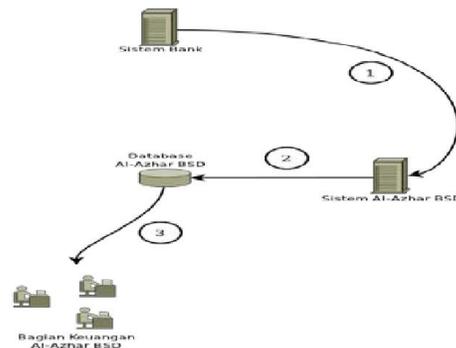
Perangkat lunak yang digunakan untuk mengembangkan sistem ini di antaranya adalah Ubuntu versi *desktop*, PHP *Hypertext Preprocessor* (PHP), MySQL *Community Edition*, Apache HTTP *Server*, dan XAMPP for Linux. Rincian perangkat lunak yang digunakan dapat dilihat pada Tabel 2..

Tabel 8. Spesifikasi Perangkat Lunak

Ubuntu	11.10 (Oneiric)
PHP	5.5.9
MySQL <i>Community Edition</i>	5.6.16
Apache HTTP <i>Server</i>	2.4.7
XAMPP for Linux	1.8.3-3

IV. PEMBAHASAN

4.1 Sistem Berjalan



Gambar 11. Proses pada Sistem Berjalan

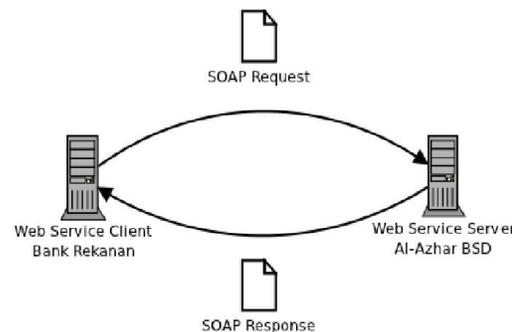
Gambar 1 memperlihatkan gambaran dalam lingkup global dari proses yang terjadi pada saat transaksi. Berikut ini adalah penjelasannya.

1. Data pembayaran dikirim dari sistem bank ke sistem Al-Azhar BSD melalui *web service*.
2. Data pembayaran disimpan di *database* keuangan Al-Azhar BSD.
3. Kasir bagian keuangan melihat catatan pembayaran harian yang ada pada *database* melalui aplikasi kasir sebagai masukan untuk membuat rekap harian pembayaran.

4.1.1 Komunikasi Sistem Al-Azhar BSD dengan Sistem Bank Rekanan

Peran *WS client* sistem bank adalah memberitahukan sistem Al-Azhar BSD setiap ada pembayaran melalui metode *online* yang masuk ke rekening Al-Azhar BSD dengan memanggil fungsi *payment*. Selain itu, *WS client* sistem bank juga dapat meminta informasi tagihan dari sistem Al-Azhar BSD dengan memanggil fungsi *inquiry*.

Peran *WS server* Al-Azhar BSD adalah menerima data pembayaran yang dikirimkan oleh sistem bank. Selain itu, *WS server* Al-Azhar BSD juga memberikan informasi tagihan jika sistem bank memintanya.



Gambar 12. Komunikasi Sistem Al-Azhar BSD dengan Sistem

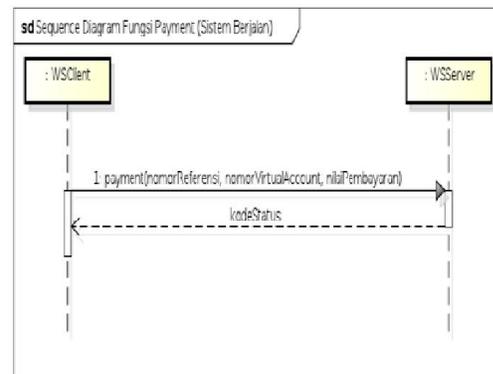
Komunikasi antara sistem Al-Azhar BSD dengan sistem bank rekanan dilakukan oleh WS

server Al-Azhar BSD dengan *WS client* bank rekanan. *WS client* mengirim *SOAP request* ke *WS server*, kemudian *WS server* menerimanya dan mengirimkan *SOAP response* ke *WS client*.

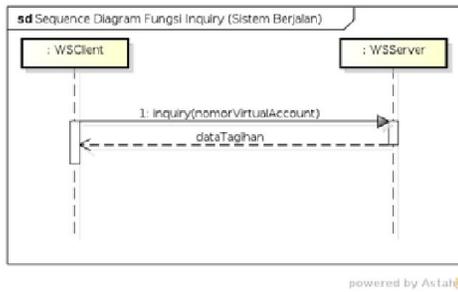
SOAP request berisi pemanggilan suatu fungsi yang ada pada *WS server* beserta argumen-argumen masukan yang digunakan dalam pemanggilan fungsi itu. *WS server* menjalankan fungsi yang diminta, kemudian mengirimkan *SOAP response* yang berisi hasil pemanggilan fungsi tadi ke *WS client*. Tabel 3 menunjukkan fungsi-fungsi yang ada pada *WS server*.

Tabel 3. Fungsi-Fungsi pada *Web Service Server*

Fungsi	Masukan	Keluaran	Deskripsi
<i>Payment</i>	Nomor referensi, nomor <i>virtual account</i> , dan nilai pembayaran.	Kode status: 00 (sukses) 12 (transaksi tidak valid) 13 (nilai tidak valid) 15 (tagihan tidak ada) 16 (nilai penuh tidak valid) 30 (format salah)	Fungsi untuk mengirim data pembayaran.
<i>Inquiry</i>	Nomor <i>virtual account</i> .	Data tagihan yang terdiri dari nama siswa dan nilai tagihan.	Fungsi untuk memperoleh data tagihan.



Gambar 13. *Sequence Diagram* Fungsi *Payment* (Sistem



Gambar 14. Sequence Diagram Fungsi Inquiry (Sistem Berjalan)

4.1.2 Proses Pembayaran Online

Proses pembayaran online dimulai dari pembayar yang menyetor sejumlah uang ke virtual account miliknya kemudian sistem bank akan mengirimkan data setoran tersebut sebagai data pembayaran ke web service Al-Azhar BSD. Web service Al-Azhar BSD akan memproses data pembayaran dan menyimpannya di database jika data pembayaran valid atau menolak pembayaran jika tidak valid. Setelah itu, web service Al-Azhar akan mengirimkan status pembayaran ke sistem bank agar sistem bank dapat menginformasikan pembayar mengenai status pembayarannya.

Aplikasi kasir mengambil data pembayaran berdasarkan tanggal yang diinginkan dari database Al-Azhar BSD untuk ditampilkan kepada kasir keuangan. Kasir keuangan akan membuat rekap harian dari data transaksi yang ditampilkan aplikasi kasir.

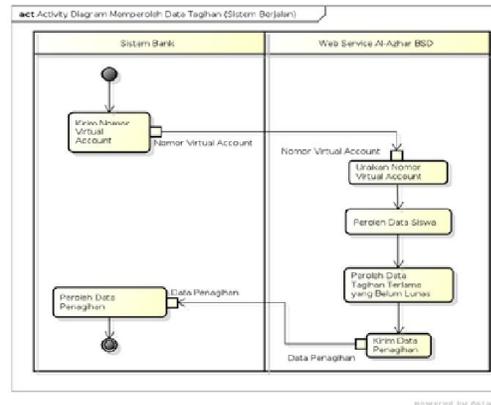
4.1.3 Use Case Diagram Sistem Pembayaran (Sistem Berjalan)

Gambar 5 memperlihatkan apa saja yang digunakan pada sistem pembayaran yang sedang berjalan. Penjelasan masing-masing use case apa pada Tabel 4.

Tabel 49. Daftar Use Case Sistem Pembayaran (Sistem Berjalan)

No.	Use Case	Deskripsi	Aktor
1.	Memperoleh Data Tagihan	Mengambil data tagihan dari web service Al-Azhar BSD.	Sistem Bank
2.	Mengirim Data Pembayaran	Mengirim data pembayaran ke web service Al-Azhar BSD.	Sistem Bank
3.	Melihat Daftar Transaksi	Melihat transaksi-transaksi pembayaran yang masuk ke rekening Al-Azhar BSD.	Kasir Keuangan
4.	Memperoleh Data Transaksi	Mengambil data transaksi dari database web service Al-Azhar BSD.	Aplikasi Kasir

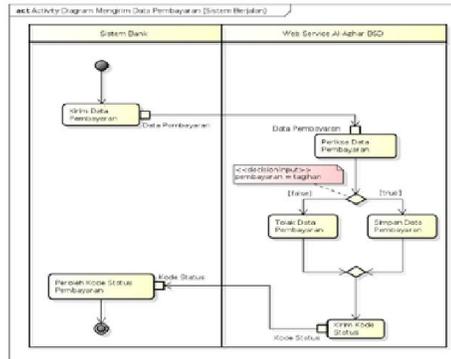
4.1.4 Activity Diagram Memperoleh Data Tagihan (Sistem Berjalan)



Gambar 6. Activity Diagram Memperoleh Data Tagihan (Sistem Berjalan)

Pada activity diagram “memperoleh data tagihan”, sistem bank mengirimkan nomor virtual account ke web service Al-Azhar BSD. Web service mengambil data tagihan berdasarkan nomor virtual account yang dikirimkan sistem bank. Data tagihan yang diperoleh dikirimkan ke sistem bank.

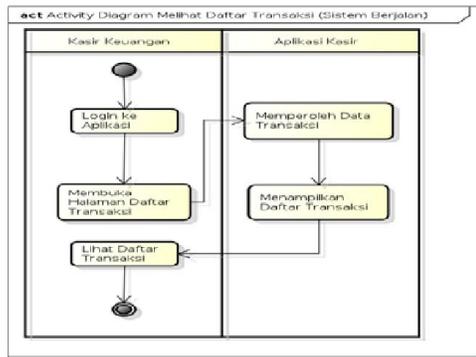
4.1.5 Activity Diagram Mengirim Data Pembayaran (Sistem Berjalan)



Gambar 7. Activity Diagram Mengirim Data Pembayaran (Sistem Berjalan)

Pada *activity diagram* “mengirim data pembayaran”, sistem bank mengirimkan data pembayaran yang terdiri dari nomor *virtual account* dan nilai pembayaran ke *web service* Al-Azhar BSD. *Web service* memproses data pembayaran kemudian mengirimkan kode status pembayaran ke sistem bank.

4.1.6 Activity Diagram Melihat Daftar Transaksi (Sistem Berjalan)

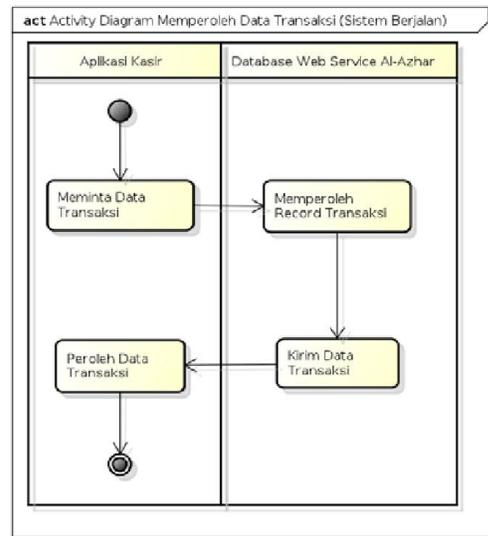


Gambar 8. Activity Diagram Melihat Daftar Transaksi (Sistem Berjalan)

Pada *activity diagram* “melihat daftar transaksi”, kasir keuangan hendak melihat daftar transaksi pembayaran yang masuk ke rekening Al-Azhar BSD. Kasir keuangan *login* ke aplikasi agar bisa membuka halaman daftar transaksi. Kasir membuka halaman daftar transaksi kemudian

aplikasi kasir akan memperoleh data transaksi dari *database*. Setelah itu aplikasi kasir akan menampilkan daftar transaksi kepada kasir keuangan.

4.1.7 Activity Diagram Memperoleh Data Transaksi (Sistem Berjalan)



Gambar 9. Activity Diagram Memperoleh Data Transaksi (Sistem Berjalan)

Pada *activity diagram* “memperoleh data transaksi”, aplikasi kasir mengambil data transaksi dari *database web service* Al-Azhar BSD. *Database* mengambil *record* transaksi, kemudian mengirimkan *record* transaksi ke aplikasi kasir.

4.2. Identifikasi Masalah

Customer bersama penulis mendiskusikan sistem berjalan untuk mengidentifikasi masalah-masalah yang ada. Tabel III.3 memaparkan masalah-masalah yang ditemukan beserta solusi untuk menyelesaikan masalah-masalah tersebut.

Tabel 5. Identifikasi Masalah Sistem Berjalan

Subjek	Sistem Berjalan	Masalah	Solusi
Transaksi Pembayaran	Melakukan transaksi sebanyak	Terlalu banyak transaksi jika	Memungkinkan pembayaran

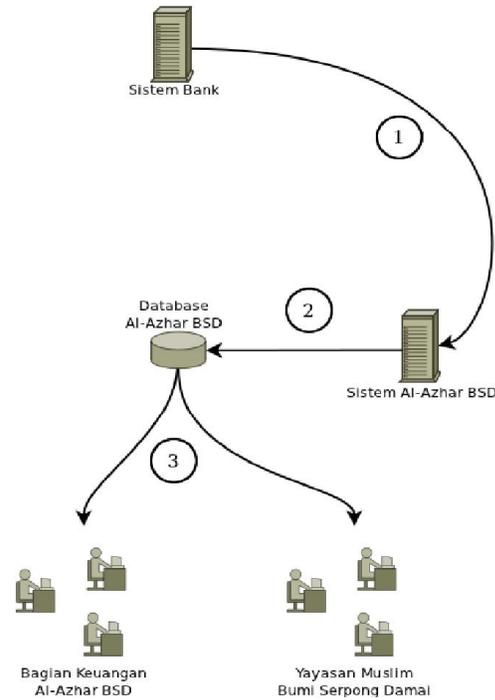
Subjek	Sistem Berjalan	Masalah	Solusi
	tagihan yang ada.	tagihan ada banyak.	n beberapa tagihan cukup dengan satu kali transaksi.
Penguraian Nomor <i>Virtual Account</i>	Mengurai-kan nomor <i>virtual account</i> menjadi kode jenis pembayaran dan NIS.	NIS yang diperoleh dari penguraian nomor <i>virtual account</i> tidak akurat untuk NIS 8 angka dan NIS sebelum angkatan tahun 2010/2011.	Memper-baiki algoritma penguraian nomor <i>virtual account</i> .
<i>Inquiry</i>	Menampil-kan nilai tagihan bulan paling awal yang belum terbayar.	Tidak menampil-kan jumlah nilai tagihan hingga bulan berjalan.	Menjumlah-kan nilai tagihan yang belum dilunasi mulai dari tagihan bulan paling awal hingga tagihan bulan berjalan.
Laporan Pembayaran	Melaporkan <i>log</i> transaksi.	Sama saja seperti melihat rekening koran. Kasir bagian keuangan memerlukan informasi rekap harian pembaya-ran.	Mengolah <i>log</i> pembaya-ran menjadi informasi rekap harian pembaya-ran.
Pemantau Perkembangan Pembayaran	Belum ada.	Diperlukan oleh yayasan dan bagian keuangan.	Membuat modul pemantau perkemban-gan pembaya-ran.

4.3 Sistem Usulan

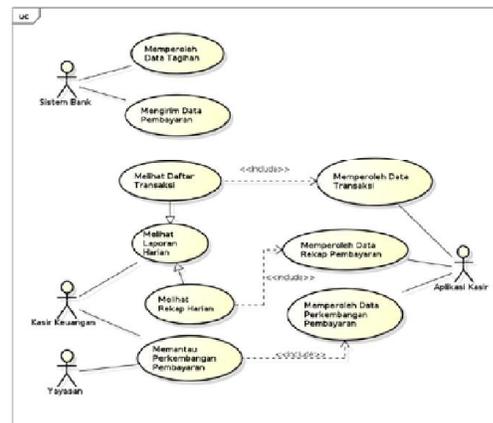
Sistem yang dikembangkan memiliki perbedaan yang tidak terlalu banyak pada lingkup global atau tingkat yang lebih tinggi jika dibandingkan dengan sistem berjalan. Gambar 10 menggambarkan proses pembayaran pada sistem yang dikembangkan. Apabila dibandingkan dengan

proses pada sistem berjalan, hanya ada penambahan akses informasi kepada yayasan.

Pada tingkat yang lebih rendah, perubahan dilakukan pada aktivitas-aktivitas yang terkait dengan pembayaran, laporan harian, dan informasi penagihan.



Gambar 10. Proses pada Sistem yang Dikembangkan



Gambar 11. Use Case Diagram Sistem Pembayaran (Sistem Usulan)

Use case diagram pada Gambar 11 menunjukkan penggunaan sistem pembayaran yang

telah dikembangkan. Terlihat adanya penambahan 5 *use case* baru yaitu:

1. melihat laporan harian yang merupakan generalisasi dari melihat daftar transaksi dan melihat rekap harian,
2. melihat rekap harian,
3. memantau perkembangan pembayaran,
4. memperoleh data rekap pembayaran, dan
5. memperoleh data perkembangan pembayaran.

Selain 5 *use case* tambahan yang telah disebutkan, ada 8 *use case* tambahan lagi yang merupakan spesialisasi dari *use case* “melihat rekap harian” dan “memantau perkembangan pembayaran”. 8 *use case* ini adalah:

1. melihat rekap harian unit,
2. melihat rekap harian kasir,
3. melihat rekap harian rekening,
4. memantau perkembangan pembayaran keseluruhan unit,
5. memantau perkembangan pembayaran unit,
6. memantau perkembangan pembayaran kelas,
7. memantau perkembangan pembayaran rombongan belajar, dan
8. memantau perkembangan pembayaran siswa.

Deskripsi *use case* baru yang ditambahkan terdapat pada sub bab Pengembangan Sistem.

4.4 Pengembangan Sistem

4.4.1 Perencanaan

Ada tiga *story* yang direncanakan untuk diimplementasikan seperti yang ditunjukkan oleh Tabel 6. *Story* diperkirakan waktu implementasinya dan diurutkan dari prioritas paling tinggi hingga paling rendah.

Tabel 6. Perencanaan *Story*

No.	Story	Perkiraan (Hari)
1.	Nilai transaksi pembayaran bebas.	5
2.	Rekap harian pembayaran.	5

No.	Story	Perkiraan (Hari)
3.	Pemantauan perkembangan pembayaran.	5

4.4.2 Analisis

Penulis menanyakan kebutuhab setiap *story* kepada *customer* kemudian mendaftarkan *requirement* setiap *story* seperti pada Tabel 7.

Tabel 7. Analisis Kebutuhan

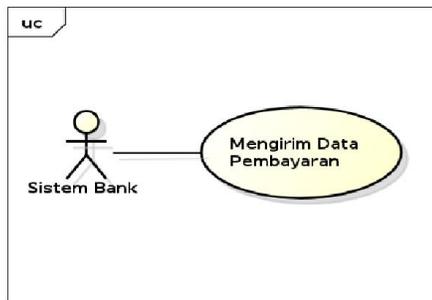
No.	Story	Requirement
1.	Nilai transaksi pembayaran bebas.	1. Bisa membayar kurang dari nilai tagihan. 2. Bisa membayar lebih dari nilai tagihan. 3. Beberapa tagihan dapat dibayarkan dengan satu transaksi. 4. Penguraian nomor <i>virtual account</i> . a) Nomor <i>virtual account</i> memiliki panjang 16 digit. b) Angka ke 1 – 4 adalah kode lembaga rekanan bank. c) Angka ke 5 adalah identitas unit. d) Angka ke 6 adalah jenis pembayaran. e) Angka ke 7 – 16 adalah NIS. f) NIS yang panjangnya kurang dari 10 angka ditambahkan angka nol di sisi kiri hingga panjang seluruhnya mencapai 10 angka. 5. Jumlah nilai tagihan adalah akumulasi sisa tagihan yang belum dibayar hingga waktu <i>inquiry</i> .
2.	Rekap harian pembayaran.	1. Rekap harian unit. 2. Rekap harian rekening. 3. Rekap harian kasir.
3.	Pemantauan perkembangan pembayaran.	1. <i>Progress</i> keseluruhan. 2. <i>Progress</i> unit. 3. <i>Progress</i> kelas. 4. <i>Progress</i> paralel. 5. <i>Progress</i> siswa.

4.4.3 Perancangan

1. Use Case Diagram

Tabel 8. Daftar Use Case Pembayaran Bebas

No.	Use Case	Deskripsi	Aktor
1	Mengirim Data Pembayaran	Mengirim data pembayaran ke <i>web service</i> Al-Azhar BSD.	Sistem Bank

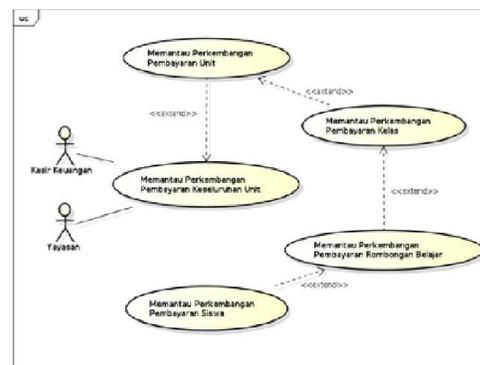


Gambar 12. Use Case Diagram Pembayaran Bebas

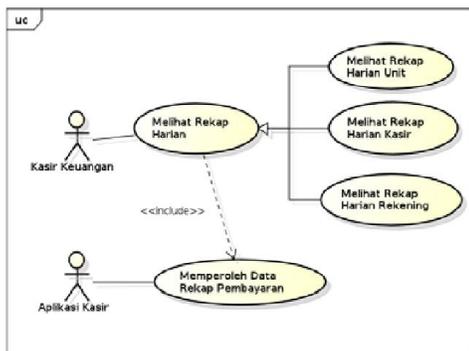
No.	Use Case	Deskripsi	Aktor
		tertentu.	
3.	Memantau Perkembangan Pembayaran Kelas	Memantau perkembangan pembayaran kelas tertentu.	Kasir Keuangan, Yayasan
4.	Memantau Perkembangan Pembayaran Rombongan Belajar	Memantau perkembangan pembayaran rombongan belajar tertentu.	Kasir Keuangan, Yayasan
5.	Memantau Perkembangan Pembayaran Siswa	Memantau perkembangan pembayaran siswa tertentu.	Kasir Keuangan, Yayasan

Tabel 9. Daftar Use Case Rekap Harian

No.	Use Case	Deskripsi	Aktor
1.	Melihat Rekap Harian Unit	Melihat rekap harian setiap unit.	Kasir Keuangan
2.	Melihat Rekap Harian Kasir	Melihat rekap harian setiap kasir.	Kasir Keuangan
3.	Melihat Rekap Harian Rekening	Melihat rekap harian setiap rekening.	Kasir Keuangan
4.	Memperoleh Data Rekap Pembayaran	Mengambil data rekap pembayaran dari database web service Al-Azhar BSD.	Aplikasi Kasir



Gambar 14. Use Case Diagram Pemantau Perkembangan Pembayaran

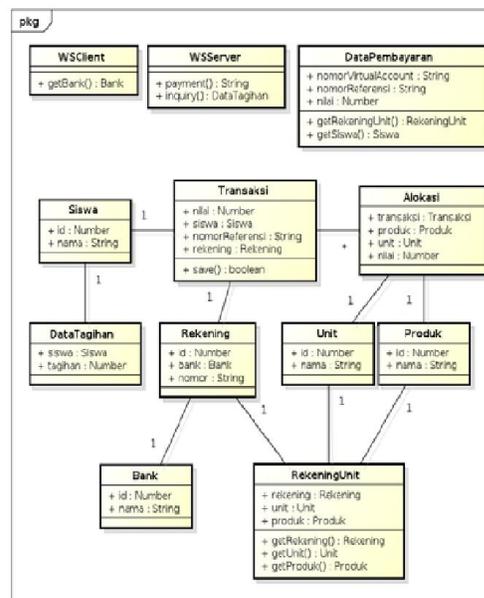


Gambar 13. Use Case Diagram Rekap Harian

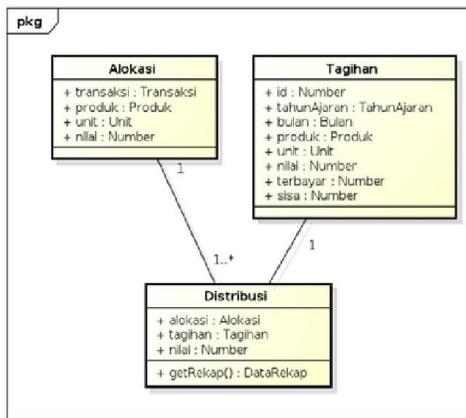
Tabel 10. Daftar Use Case Pemantau Perkembangan Pembayaran

No.	Use Case	Deskripsi	Aktor
1.	Memantau Perkembangan Pembayaran Keseluruhan Unit	Memantau perkembangan pembayaran seluruh unit.	Kasir Keuangan, Yayasan
2.	Memantau Perkembangan Pembayaran Unit	Memantau perkembangan pembayaran unit	Kasir Keuangan, Yayasan

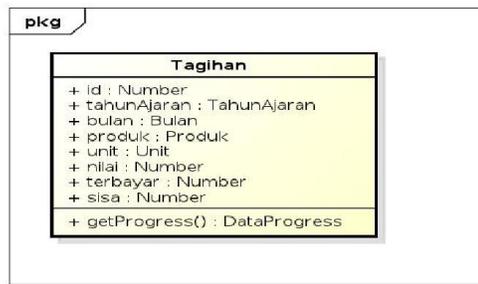
2. Class Diagram



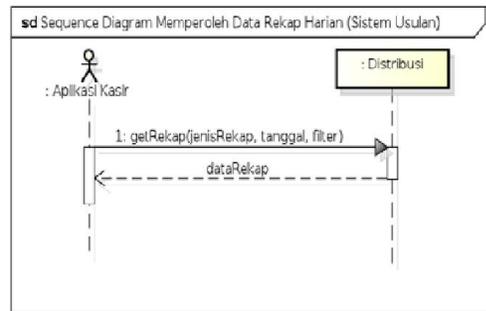
Gambar 15. Class Diagram Mengirim Data Pembayaran



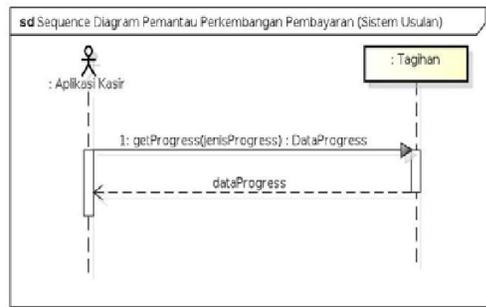
Gambar 16. Class Diagram Memperoleh Data Rekap Harian



Gambar 17. Class Diagram Pemantau Perkembangan Pembayaran

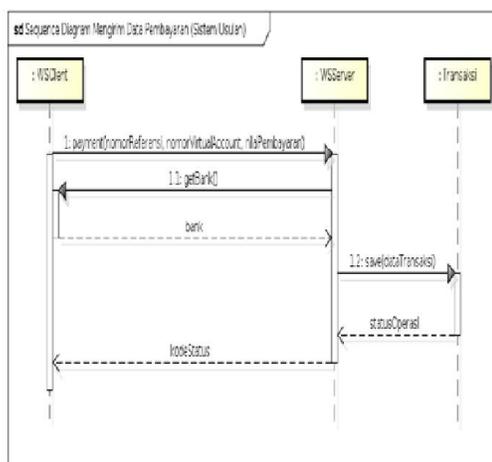


Gambar 19. Sequence Diagram Memperoleh Data Rekap Harian



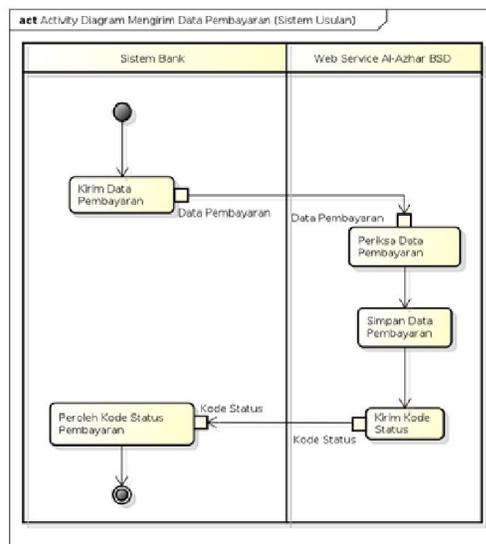
Gambar 20. Sequence Diagram Pemantau Perkembangan Pembayaran

3. Sequence Diagram

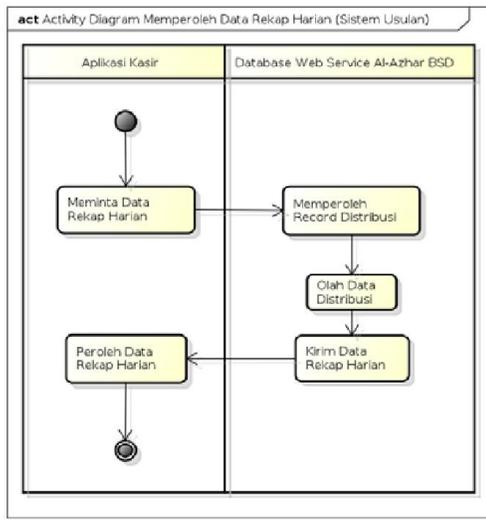


Gambar 18. Sequence Diagram Mengirim Data Pembayaran

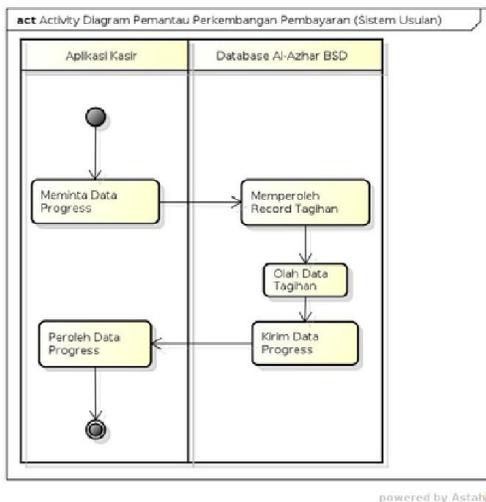
4. Activity Diagram



Gambar 21. Activity Diagram Mengirim Data Pembayaran



Gambar 22. Activity Diagram Memperoleh Data Rekap Harian



Gambar 23. Activity Diagram Pemantau Perkembangan Pembayaran

1. Coding

Coding menggunakan bahasa pemrograman PHP.

2. Pengujian

Pengujian dilakukan dengan *pilot testing*. Semua pengujian dinyatakan sukses.

3. Pengiriman

Aplikasi yang telah dibangun dan diuji dikirimkan kepada *customer* melalui surat elektronik.

4. Iterasi

Ada 3 iterasi pada pengembangan sistem ini. Iterasi terjadi pada tahap analisis, perancangan, coding, dan pengujian.

a. Iterasi Pertama

Sistem dijadwalkan untuk rilis dalam waktu 1 pekan. *Story* pertama dipilih untuk diimplementasikan. Perkiraan waktu implementasi adalah 5 hari sehingga rilis berikutnya hanya cukup untuk implementasi *story* pertama.

b. Iterasi Kedua

Sistem dijadwalkan untuk rilis dalam waktu 1 pekan. *Story* kedua dipilih untuk diimplementasikan. Perkiraan waktu implementasi adalah 5 hari sehingga rilis berikutnya hanya cukup untuk implementasi *story* kedua tanpa mengubah hasil implementasi *story* pertama yang sudah ada.

c. Iterasi Ketiga

Sistem dijadwalkan untuk rilis dalam waktu 1 pekan. *Story* ketiga dipilih untuk diimplementasikan. Perkiraan waktu implementasi adalah 5 hari sehingga rilis berikutnya hanya cukup untuk implementasi *story* ketiga tanpa mengubah hasil implementasi *story* pertama dan kedua yang sudah ada.

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. Pembayaran fleksibel diterapkan dengan membuat transaksi pembayaran tidak terikat pada nilai tagihan saat inquiry sehingga siswa atau orangtua/wali dapat melakukan pembayaran dengan nilai berapa saja (lebih besar atau sama dengan nilai minimum transfer) dan dapat membayar beberapa tagihan dengan 1 (satu) transaksi pembayaran.

2. Sistem mencatat distribusi alokasi transaksi ke tagihan-tagihan yang ada untuk menghasilkan rekap harian yang dapat digunakan oleh kasir bagian keuangan sehingga kasir dapat memperoleh informasi rekap harian tanpa mengolah data pembayaran secara manual karena telah diolah secara otomatis oleh sistem.
 3. Sistem mengolah data tagihan yang terdiri dari nilai tagihan dan nilai yang telah terbayar untuk menyajikan informasi perkembangan pembayaran sehingga bagian keuangan dan yayasan dapat melihat perkembangan pembayaran melalui sistem yang telah dikembangkan. type.
- 5.2. Saran-Saran
1. Untuk mengantisipasi pengolahan data besar.
 2. Rapikan rancangan kode sumber aplikasi agar lebih mudah dimengerti dan dipelihara.
3. Kembangkan unit test dan integration test untuk memastikan tidak ada kesalahan pada setiap unit aplikasi dan integrasi antar unit aplikasi.
 4. Kembangkan modul pencetakan rekap harian pembayaran karena sistem yang dikembangkan masih mengandalkan peramban untuk menangani pencetakan.

REFERENSI

- [1] Box, Don; Ehnebushe, David; Kakivaya, Gopal; Layman, Andrew; Mendelshon, Noah; Nielson, Henrik; Thatte, Satish; dan Winner, Dave. 2000. Simple Object Access Protocol (SOAP) 1.1 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508> (Diakses pada 3 Oktober 2013).
- [2] Christensen, Erik; Curbera, Francisco; Meredith, Greg; Weerawarana, Sanjiva; 2001. Web Services Description Language (WSDL) 1.1 <http://www.w3.org/TR/2001/NOTE-WSDL-20010315> (diakses pada 3 Oktober 2013)