



Perancangan Aplikasi Saldo pada Kartu Santri di Pondok Pesantren Melalui Transfer Bank

Abd. Wahab Syahroni¹, Nilam Ramadhani², Badar Said³

^{1,2,3}Program Studi Informatika, Fakultas Teknik, Universitas Madura, Jl. Raya Panglegur No.Km 3,5 Kec. Tlanakan, Kab. Pamekasan, Jawa Timur 69371, Indonesia

ABSTRACT

This research is a continuation of previous research, the shortcoming in previous research is that there is no additional balance on the santri card via bank transfer. To add balance, the Guardian Santri must visit the Islamic Boarding School. Due to the corona virus period, the caretaker of the Islamic boarding school wants the current application so that it can be developed, how can the Guardian Santri increase the student card balance via bank transfer. The purpose of this study is to develop an application to add balance to the santri card at the Islamic boarding school so that it can be added to the balance via bank transfer so that the guardians of the santri no longer need to visit the Islamic boarding school if they only want money to the santri. The Rapid Application Development method is used in building this application because it has advantages in application development. Use case diagrams are used to describe the interaction between the processes in the application and the actors involved, making it easier to see the flow of the application being built. The application has been successfully developed based on the website, based on application testing use black box method, all features have been able to run well, especially the virtual account creation feature and adding balance to the santri card.

Keywords: Islamic boarding school, Student Card, Application, Corona Virus, Bank transfer

ABSTRAK

Penelitian ini merupakan kelanjutan dari penelitian sebelumnya, kekurangan pada penelitian sebelumnya adalah tidak adanya fitur penambahan saldo pada kartu santri melalui transfer bank. Untuk melakukan tambah saldo, Wali Santri harus berkunjung ke Pondok Pesantren. Dikarenakan masih masa virus corona, Pengasuh Pondok Pesantren menginginkan aplikasi yang ada saat ini agar dapat dilakukan pengembangan, bagaimana caranya agar Wali Santri dapat menambah saldo kartu santri melalui transfer bank. Tujuan penelitian ini untuk mengembangkan aplikasi tambah saldo pada kartu santri di pondok pesantren agar dapat dilakukan penambahan saldo melalui transfer bank sehingga wali santri tidak perlu lagi berkunjung ke pondok pesantren jika hanya ingin memberikan uang kepada santri. Metode Rapid Application Development digunakan dalam membangun aplikasi ini karena memiliki keuntungan dalam kecepatan pengembangan aplikasi. Use case diagram digunakan dalam menggambarkan interaksi antara proses yang ada dalam aplikasi dengan aktor yang terlibat sehingga memudahkan melihat alur penggunaan aplikasi yang sedang dibangun. Aplikasi telah berhasil dikembangkan dengan berbasis web, berdasarkan hasil pengujian aplikasi menggunakan metode black box testing, semua fitur telah dapat berjalan dengan baik terutama fitur pembuatan virtual account dan fitur penambahan saldo pada kartu santri.

Kata Kunci: Pondok Pesantren, Kartu Santri, Aplikasi, Virus Corona, Transfer Bank

1. PENDAHULUAN

Dari waktu ke waktu, kemajuan teknologi terus berkembang mulai dari teknologi informasi dan komunikasi, teknologi pertanian, dan teknologi industri, dimana perkembangan teknologi ini berdampak pada kehidupan bermasyarakat, sehingga setiap orang tertarik memanfaatkan setiap perkembangan ini salah satunya adalah penggunaan digital money. Pada era ini, model transaksi dalam bisnis telah berubah dimana penggunaan uang digital menjadikan proses transaksi semakin cepat, efektif, efisien dan penuh kemudahan [1]. Perkembangan internet dan teknologi informasi telah memacu tuntutan dan kebutuhan perkembangan teknologi di Pondok Pesantren, terlebih pada situasi pandemi virus corona ini dimana masyarakat harus melakukan jaga jarak. Pengintegrasian inovasi dalam internet dan teknologi informasi akan memunculkan teknologi baru yang dapat bermanfaat untuk Pondok Pesantren terutama untuk menghindari kontak langsung antara lingkungan Pondok Pesantren dengan lingkungan luar. Pondok Pesantren menghadapi tantangan supaya dapat memenuhi tuntutan konektivitas antara para Santri dengan Wali Murid pada situasi virus corona saat ini yang tidak dapat ditentukan kapan akan berakhir.

Pada penelitian sebelumnya[2], pernah membuat sebuah aplikasi e-money di Pondok Pesantren Puncak Darussalam yang disebut dengan aplikasi e-nuood, inti dari aplikasi tersebut adalah bagaimana agar santri tidak menggunakan uang tunai dalam setiap transaksi keuangan yang ada di Pondok Pesantren Puncak Darussalam namun cukup menggunakan kartu santri. Aplikasi tersebut masih digunakan sampai saat ini. kekurangan dari aplikasi tersebut adalah tidak adanya fitur penambahan saldo kartu santri melalui transfer bank. Untuk melakukan tambah saldo, Wali Santri harus berkunjung ke Pondok Pesantren. Dikarenakan masih masa virus corona, Pengasuh Pondok Pesantren menginginkan agar aplikasi yang ada saat ini agar dapat dilakukan pengembangan bagaimana caranya agar Wali Santri dapat menambah saldo kartu santri melalui transfer bank.

Sebagaimana kita ketahui bahwasannya saat ini sudah banyak sistem yang dapat terhubung dengan sistem pembayaran di bank seperti terhubung secara host to host, server ke server. Maka berangkat dari alasan ini, penulis akan melanjutkan penelitian sebelumnya agar dapat memanfaatkan teknologi Internet of Things dan Application Programming Interface untuk di implementasikan pada Pondok Pesantren sehingga kartu santri yang ada di Pondok Pesantren dapat ditambah saldonya melalui transfer bank.

Internet of Thing atau lebih dikenal dengan istilah "IoT" adalah sebuah konsep yang memanfaatkan konektivitas internet agar dapat menghubungkan peralatan, mesin maupun perangkat lainnya untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen[3]. Application Programming Interface (API) merupakan sekumpulan fungsi, subroutine, protokol komunikasi, atau tools untuk membuat perangkat lunak[4], dengan memanfaatkan teknologi API ini maka nantinya komunikasi antar sistem yang berbeda dapat dilakukan[5].

Sebagai tambahan informasi dan kondisi saat ini di Pondok Pesantren, pertama, aplikasi e-nuqood yang ada di Pondok Pesantren saat ini berjalan pada jaringan lokal Pondok Pesantren, kedua, harus bekerja sama dengan bank tertentu. Oleh sebab itu, penelitian ini sangat menarik untuk dilakukan.

2. TINJAUAN PUSTAKA

2.1. *Internet of Things (IoT)*

IoT bekerja dengan memanfaatkan instruksi pemrograman yang setiap perintahnya bisa menghasilkan interaksi ke sesama perangkat yang terhubung secara otomatis tanpa adanya intervensi pengguna, bahkan dalam jarak jauh sekali pun. Adapun faktor vital yang menjadi kelancaran perangkat IoT adalah jaringan internet yang menjadi konektor antara sistem dan perangkat. Sementara, manusia dalam tahap ini hanya menjadi monitor untuk setiap perilaku perangkat saat mereka bekerja[6]. Disini kami menggunakan teknologi Internet of Things karena nantinya wali santri dapat menggunakan ATM atau mobil banking yang membutuhkan jaringan internet untuk melakukan transfer uang ke kartu santri di Pondok Pesantren kemudian melakukan komunikasi dengan server yang ada di Pondok Pesantren.

2.2. *Application Programming Interface (API)*

API merupakan Suatu software yang dapat mengizinkan dan mengintegrasikan dua sistem atau lebih yang berbeda agar dapat saling terhubung[7]. Penggunaan API bertujuan agar saling berbagi data antar sistem yang berbeda, serta mempercepat proses pengembangan aplikasi dengan cara menyiapkan fungsi yang terpisah sehingga pengembang aplikasi tidak perlu lagi membuat fitur yang serupa.

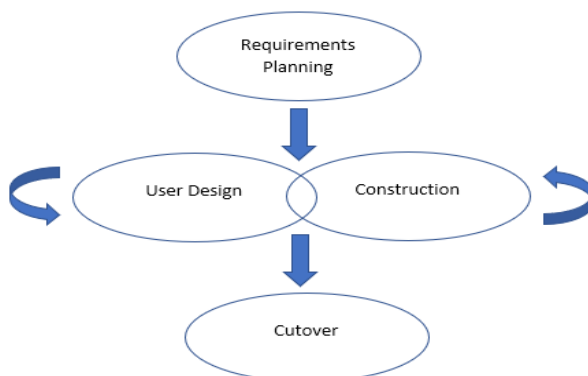
RESTAPI merupakan salah satu dari desain arsitektur yang ada dalam API. RESTAPI adalah sebuah penerapan konsep perpindahan antar state yang juga merupakan salah satu dari jenis web service. adapun gambaran state itu seperti ketika browser melakukan request suatu halaman web maka server akan merespon dan mengirimkan state halaman web yang ada saat ini ke browser. REST bekerja menggunakan navigasi link HTTP untuk perpindahan antar state tersebut. Perintah HTTP yang dapat dipakai seperti fungsi GET, POST, PUT, atau DELETE. Adapun respon yang dikirimkan dapat berupa format teks, XML atau JSON[8]. Sehingga respon yang diperoleh dapat dengan mudah dimengerti dan diolah oleh client[9]. REST merupakan teknologi pertukaran data web service saat ini yang sangat populer[10].

2.3. *Virtual Account (VA)*

Virtual Account adalah sekumpulan akun non nyata yang digunakan untuk melakukan dan menerima pembayaran atas nama rekening[11]. Penggunaan Virtual Account oleh pihak Bank digunakan untuk mendistribusikan dan pemantauan pembayaran atau tagihan secara elektronik sehingga dapat mengurangi kesalahan dalam penginputan data pembayaran customer. Sedangkan dalam penelitian ini, Virtual account merupakan nomor identifikasi yang bersifat unik bagi Santri, sehingga Wali Santri atau orang lain dapat melakukan transfer uang dari bank kepada Virtual Account Santri yang nantinya akan di hubungkan dengan kartu santri yang ada di Pondok Pesantren.

2.4. *Rapid Application Development (RAD)*

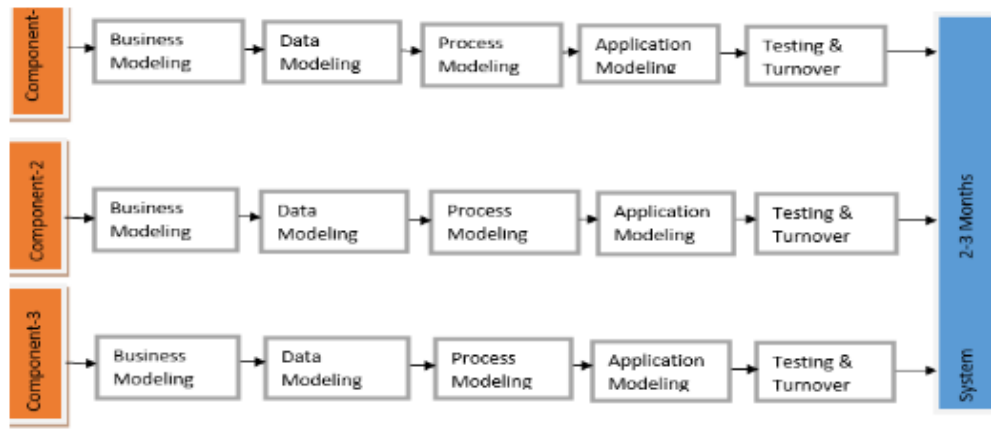
Rapid Application Development (RAD) adalah metode pengembangan perangkat lunak yang mengedepankan proses yang cepat. Dimana pengembangan RAD ini dibagi ke dalam beberapa komponen dan setiap komponen diselesaikan dalam tempo yang singkat[12].



Gambar 1. Fase RAD

1. Requirements planning merupakan proses pembuatan analisis tentang kebutuhan program dan persyaratan dasar mengenai program yang akan dibuat.
2. User design merupakan proses untuk membuat rancangan desain program yang akan dibuat berdasarkan dari kebutuhan dan persyaratan yang telah ditentukan sebelumnya.
3. Construction merupakan kolaborasi antara pengembang dan user dimana user dapat memberikan saran, masukan dan revisi terkait program yang dibuat oleh programmer atau pengembang.
4. Cutover merupakan langkah setelah melalui banyak proses dan aplikasi telah dibuat, pengembang menyerahkan aplikasi kepada pengguna.

Lima komponen RAD dapat dilihat pada gambar 2 dibawah ini.



Gambar 2. Komponen RAD [12]

1. Business Modeling merupakan tahapan pengelolaan informasi oleh pengembang aplikasi sebagai kebutuhan sistem yang terurai dalam bentuk pertanyaan.
2. Data Modelling merupakan tahap penyusunan informasi menjadi sekumpulan objek dari hasil analisa yang dilakukan oleh pengembang aplikasi pada kebutuhan sistem yang akan dibangun.
3. Process Modelling merupakan tahapan untuk mengimplementasikan masing masing fungsi pengoperasian aplikasi yang akan dibangun berdasarkan sekumpulan objek data yang telah didefinisikan sebelumnya. (penambahan, pengolahan data, modifikasi dan hapus data).
4. Application Generation merupakan proses yang memungkinkan RAD untuk melakukan proses REUSE yaitu proses penggunaan kembali komponen aplikasi yang telah dibuat sebelumnya.
5. Testing and Turnover merupakan proses uji coba program yang telah dibuat sebelum akan digunakan kembali pada komponen selanjutnya.

2.5. Perangkat Lunak Pendukung

Sistem dibangun menggunakan pemrograman web, adapun teknologi pendukung untuk pengembangan sistem ini terdiri dari postman, browser, Javascript dengan library jQuery, PHP serta MySQL sebagai basis datanya.

3. METODOLOGI PENELITIAN

3.1. Analisis Sistem

Penelitian ini merupakan penelitian lanjutan dari penelitian sebelumnya disini[2] dimana metode pengembangan aplikasi menggunakan metode prototyping, karena pada saat itu, aplikasi belum ada. Pada pengembangan aplikasi saldo melalui transfer bank ini, peneliti menggunakan metode pengembangan aplikasi Rapid Application Development (RAD) karena pada metode pengembangan perangkat lunak RAD ini lebih mengedepankan proses yang cepat dengan penambahan komponen komponen yang akan dibuat[12]. Pada penelitian ini membahas tentang aplikasi saldo pada kartu santri di Pondok Pesantren agar dapat dilakukan pengiriman uang melalui transfer Bank dengan platform pemrograman web yang dapat berjalan secara semi online dikarenakan pada penelitian sebelumnya, permasalahan ini belum bisa terselesaikan. Dikatakan semi online, karena untuk mendapatkan data dari Bank, komputer aplikasi e-nuqood harus terhubung ke internet untuk dapat mengakses API Bank. Sedangkan aplikasi saldo ada di server komputer Pondok Pesantren yang berjalan secara offline.

Untuk membangun aplikasi saldo ini perlu dijelaskan langkah langkah yang dilakukan oleh peneliti yakni terkait kebutuhan yang diperlukan baik secara non fungsional maupun yang fungsional seperti berikut (tahapan requirement pada metode RAD):

3.1.1. Kebutuhan Fungsional

1. Login digunakan untuk masuk kedalam aplikasi saldo ini menggunakan akun e-nuqood dengan hak akses admin.
2. Aplikasi saldo dapat membuat virtual account santri.

3. Aplikasi saldo dapat melakukan sinkronisasi data virtual account santri.
4. Aplikasi saldo dapat melakukan sinkronisasi saldo transfer bank yang dilakukan oleh wali santri.
5. Aplikasi saldo dapat merubah saldo santri yang ada di e-nuqood sesuai dengan transfer bank yang dilakukan oleh wali santri.
6. Aplikasi saldo dapat melakukan rekap laporan transfer bank.

3.1.2. Kebutuhan Non Fungsional

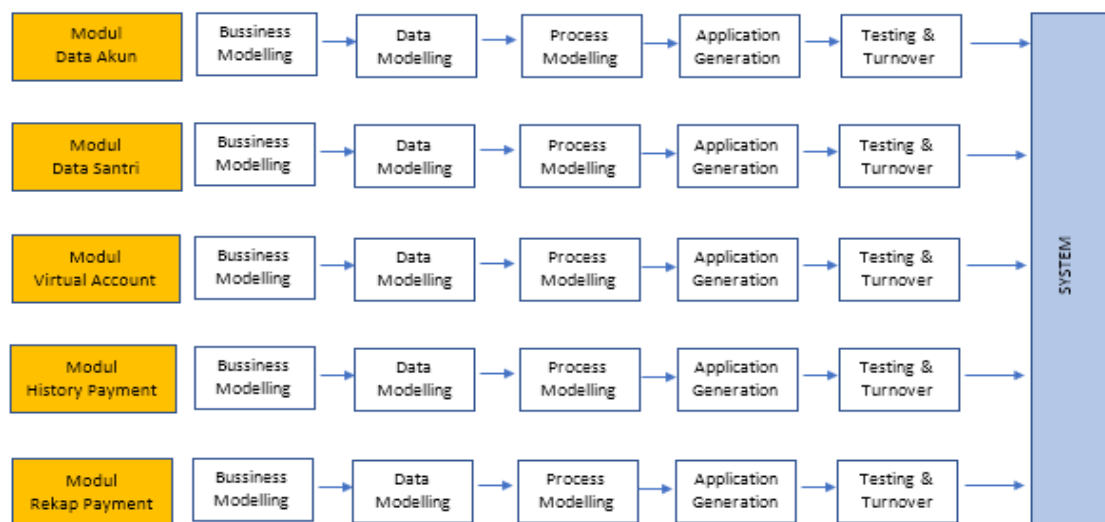
1. Aplikasi dibangun dengan pemrograman web yang diakses melalui browser pada server lokal pondok pesantren namun harus mengaktifkan jaringan internet jika ingin melakukan sinkronisasi dengan API Bank.
2. Pondok Pesantren harus bekerja sama terlebih dahulu dengan pihak Bank tertentu agar diberikan otoritas untuk mengakses API Bank.
3. Tenaga programmer dari pihak Bank akan melakukan rapat terkait aturan aturan yang dibutuhkan dalam pembuatan API.
4. HTTP Method yang digunakan antara lain POST, untuk proses pembuatan virtual account santri, GET untuk mendapatkan data virtual account santri serta history transfer Bank yang pernah dilakukan oleh wali santri pada virtual account santri.
5. Pencairan uang hasil transfer wali santri ke virtual account santri dapat dilakukan oleh Pondok Pesantren kepada Bank yang telah melakukan kerjasama.

3.2. Modul atau Fitur Aplikasi

Rancang bangun aplikasi saldo ini disusun dalam beberapa modul agar dapat mempercepat pengerjaan serta dapat dikerjakan secara terstruktur. adapun modul pengerjaan dari aplikasi saldo ini antara lain:

1. Modul menampilkan data akun dari aplikasi e-nuqood dengan hak akses admin
2. Modul menampilkan data santri dari aplikasi e-nuqood
3. Modul Virtual Account Santri, Modul ini terdiri dari pembuatan virtual account santri dan cetak virtual account santri
4. Modul History Payment, Modul ini terdiri dari sinkronisasi transfer Bank (payment) yang pernah dilakukan oleh wali santri pada virtual account santri
5. Modul Rekap Payment, Modul ini terdiri dari rekap payment per periode, rekap payment per periode per santri, total payment per periode.

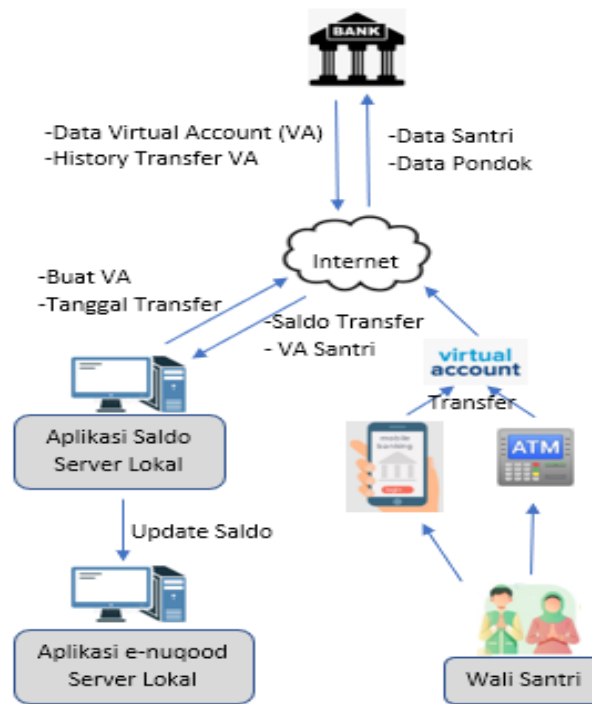
Berikut gambaran dari alur pengerjaan aplikasi saldo ini.



Gambar 3. Alur Pengerjaan Aplikasi

Langkah pertama yang dilakukan berdasarkan pada gambar 3 yakni pengumpulan kebutuhan (fitur aplikasi) sehingga dapat memenuhi kebutuhan yang diinginkan oleh pengguna, langkah kedua yakni pengumpulan data untuk membuat design database, langkah ketiga yakni proses modeling dengan menggunakan UML, langkah keempat yakni pembuatan aplikasi yang sebenarnya, dan langkah terakhir yakni pengujian aplikasi menggunakan metode black box testing.

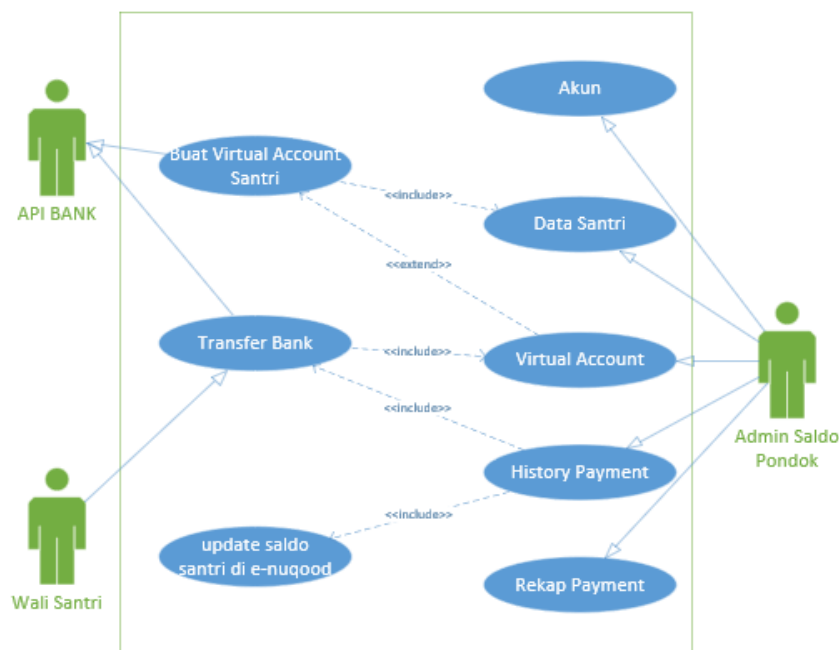
Pada gambar dibawah ini terlihat desain arsitektural dari aplikasi yang akan dibuat dimana aplikasi saldo bisa mendapatkan data dari Bank dan mengubah saldo kartu santri pada aplikasi e-nuqood (langkah user design pada metode RAD).



Gambar 4. Desain Arsitektural Sistem

3.3. Use Case Diagram

Use case diagram menggambarkan interaksi antara proses yang ada dalam aplikasi dengan aktor yang terlibat sehingga memudahkan melihat alur penggunaan aplikasi yang sedang dibangun bagi pembuat program.

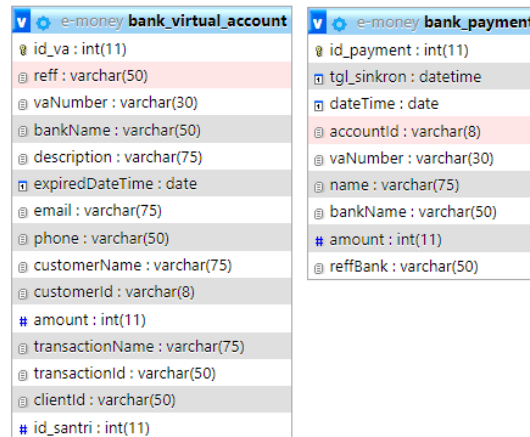


Gambar 5. Use Case Diagram Aplikasi

Dari gambar 5 tersebut terlihat bahwa admin aplikasi saldo dapat membuat virtual account santri dengan menggunakan API milik Bank dengan menyertakan data santri yang akan dibuatkan virtual account. Virtual account yang telah dibuat, dapat diberikan kepada wali santri nantinya, sehingga wali santri dapat melakukan transfer uang di Bank melalui aplikasi Bank baik berupa mobile banking maupun ATM. Setelah terdapat transfer uang dari wali santri, maka aplikasi saldo di pondok dapat melakukan sinkronisasi saldo dan melakukan update saldo santri pada aplikasi e-nuqood.

3.4. Physical Data Model (PDM)

Berikut gambar dari desain physical data model pada aplikasi saldo ini.



Gambar 6. PDM

Disini, peneliti menambahkan 2 tabel baru untuk menampung beberapa data yang dibutuhkan oleh aplikasi saldo ini yakni tabel virtual account dan tabel history payment, alasan kenapa harus ditampung pada tabel baru, agar memperkecil dependency terhadap API Bank. Tidak hanya itu, dengan ditampung di tabel baru, maka ketika aplikasi saldo berjalan secara offline, tetap dapat melihat fitur seperti virtual account dan history payment meskipun tidak bisa membuat virtual account baru dan melakukan sinkron history payment pada saat aplikasi saldo ini offline.

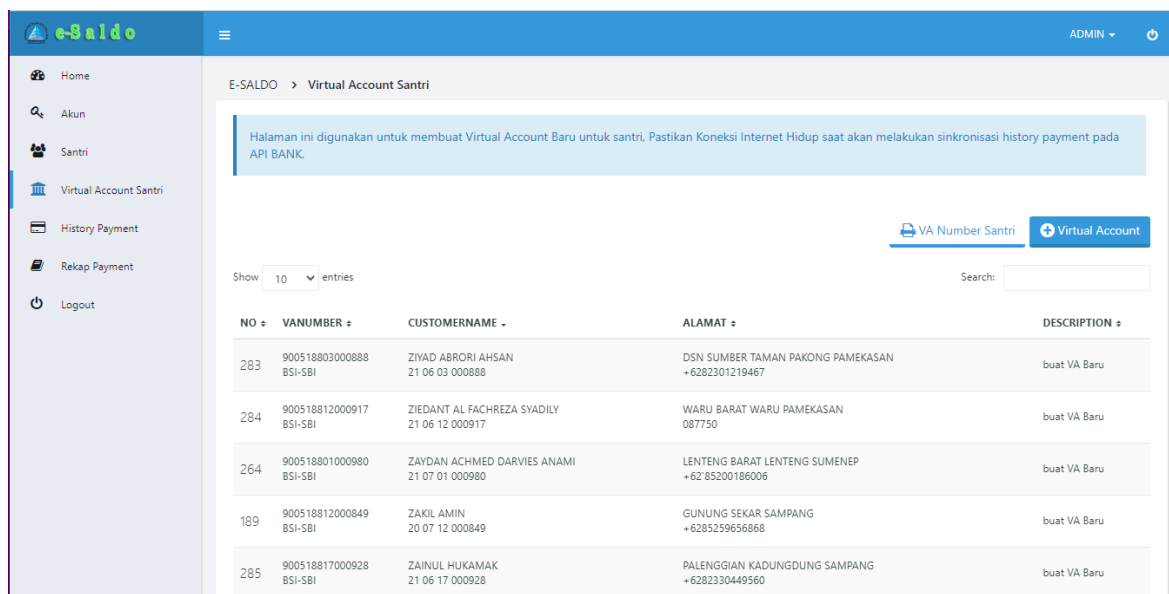
4. HASIL DAN PEMBAHASAN

4.1. Implementasi Tampilan Aplikasi

Berikut tampilan dari aplikasi saldo yang sudah selesai dibuat.

1. Tampilan Halaman Virtual Account

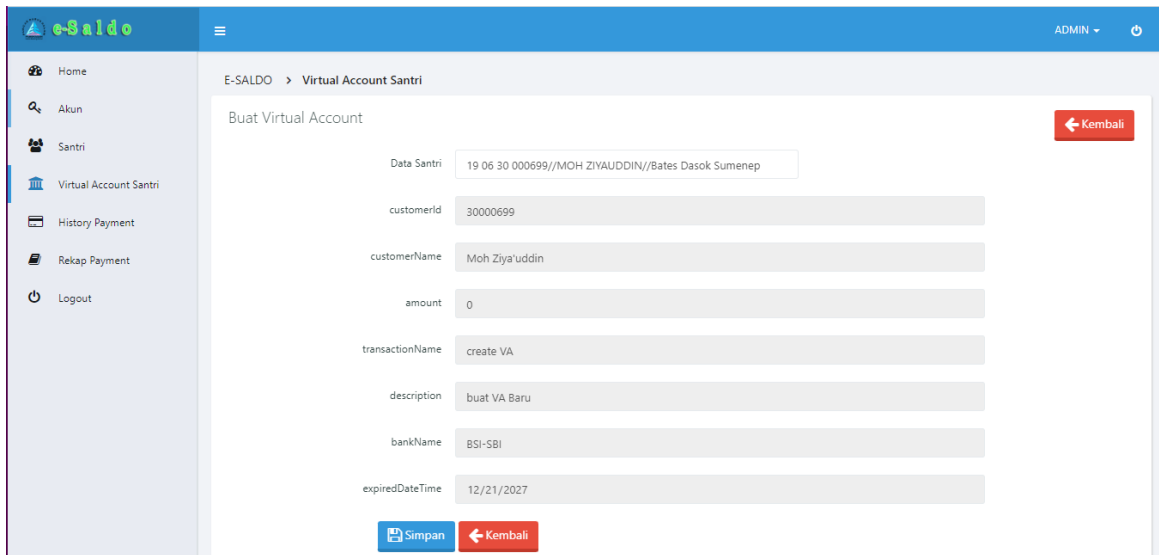
Halaman ini terdapat pada menu Virtual Account Santri, data santri yang ada pada halaman ini berarti data santri yang telah membuat virtual account. Untuk membuat virtual account baru, klik tombol warna biru bertuliskan tambah virtual account.



Gambar 7. Halaman Virtual Account

2. Tampilan Tambah Virtual Account

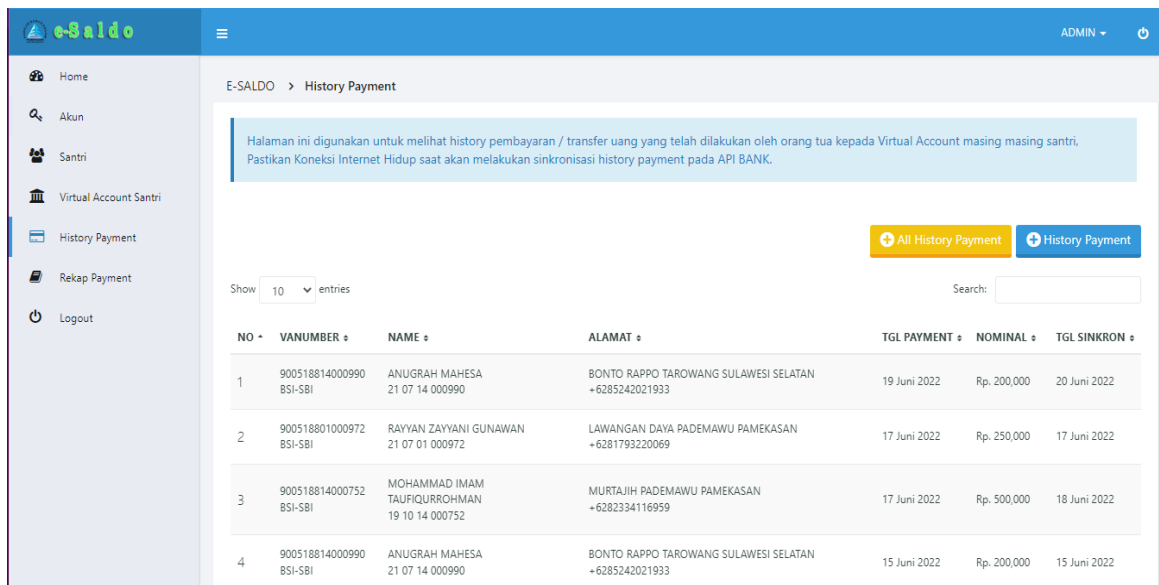
Pada menu inilah nantinya admin dapat membuat virtual account santri. admin hanya cukup mengisi data santri secara autocomplete pada form input data santri kemudian tekan tombol simpan. Saat menekan tombol simpan, data ini akan dikirimkan ke API Bank, jika berhasil, BANK akan merespon dan membuatkan virtual account santri.



Gambar 8. Buat Virtual Account

3. Tampilan Halaman History Payment

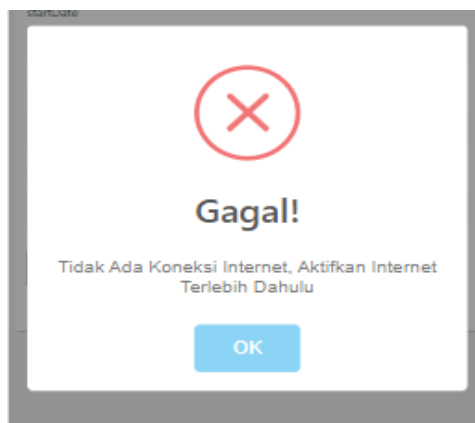
Halaman ini terletak pada menu history payment. Menu ini menampilkan data transfer ke virtual account santri. pada halaman inilah sinkron transfer bank dengan saldo santri dilakukan.



Gambar 10. History Payment

4. Tampilan Notifikasi Aplikasi

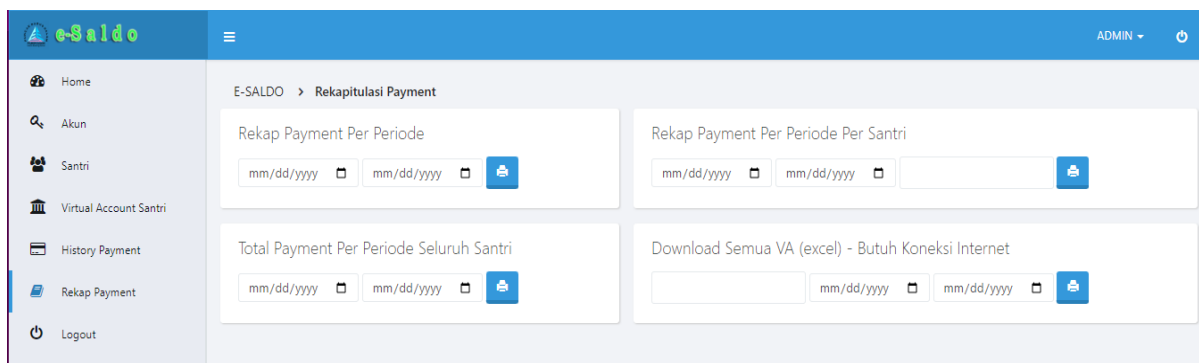
Notifikasi sangat dibutuhkan dalam setiap pembuatan aplikasi untuk memberi tahu user, respon apa yang terjadi pada aplikasi. Berikut contoh tampilan notifikasi saat aplikasi ingin mengakses internet tapi komputer tidak terhubung dengan internet.



Gambar 11. Notifikasi Internet Tidak Aktif

5. Tampilan Halaman Rekap

Halaman ini terdapat pada menu rekap payment. Pada halaman ini, admin saldo pondok dapat melakukan rekap payment.



Gambar 12. Rekap Payment

4.2. Pengujian Aplikasi

Pengujian aplikasi saldo ini menggunakan metode black box testing. Metode pengujian ini melakukan pengujian dengan melihat proses yang sedang berjalan pada aplikasi untuk dicari kesalahan atau error pada aplikasi[13]. Proses pengujian dengan metode ini dilakukan dengan melakukan skenario tertentu supaya dapat melihat kinerja dari fitur aplikasi yang telah dibuat. Berikut adalah pengujian yang dilakukan pada aplikasi saldo secara khusus pada fitur utama aplikasi ini yaitu pada saat membuat virtual account dan history payment hingga merubah jumlah saldo santri berdasarkan data transfer bank dari wali santri.

1. Pengujian pembuatan virtual account

Pengujian ini bertujuan untuk melihat apakah pembuatan virtual account telah dapat dilakukan oleh aplikasi yang telah dibuat.

Tabel 1. Pengujian Pembuatan Virtual Account

Masukan yang diberikan	Luaran yang dihasilkan	Kesimpulan
Tidak mengisi inputan NIS	Menampilkan notifikasi Data Santri harus diisi	Berhasil
NIS tidak diambil dari data santri secara autocomplete	Menampilkan notifikasi data santri harus dari data santri secara autocomplete	Berhasil
Komputer aplikasi tidak terhubung ke internet	Menampilkan notifikasi tidak ada koneksi internet, aktifkan internet terlebih dahulu	Berhasil
NIS di ambil dari autocomplete dan Komputer terhubung ke internet	Menampilkan notifikasi Virtual Account Berhasil Dibuat	Berhasil
NIS yang telah membuat virtual account dibuatkan virtual account lagi	Menampilkan notifikasi Duplicate transaction/customer ID atau Santri Ini Sudah Punya VA, Silahkan Cek Terlebih Dahulu	Berhasil

Berdasarkan tabel 1, dapat dikatakan bahwa hasil pengujian dengan skenario pembuatan virtual account telah berjalan dengan baik sesuai alternatif inputan yang telah dilakukan.

2. Pengujian history payment

Pengujian ini bertujuan untuk melihat apakah fungsionalitas fitur history payment telah dapat berjalan dengan baik.

Tabel 2. Pengujian History Payment

Masukan yang diberikan	Luaran yang dihasilkan	Kesimpulan
Melakukan transfer kepada virtual account yang telah dibuat melalui mobile banking	Bukti transfer sesuai dengan nominal yang di transfer.	Berhasil
Input NIS dari autocomplete data santri yang telah membuat virtual account	Menampilkan VA Number sesuai data santri yang dipilih	Berhasil
Komputer aplikasi tidak terhubung ke internet	Menampilkan notifikasi tidak ada koneksi internet, aktifkan internet terlebih dahulu	Berhasil
NIS, VA Number, periode tanggal sinkron dan komputer terhubung ke internet.	Menampilkan notifikasi Sinkron Payment Santri Berhasil dan saldo santri berubah pada aplikasi e-nuqood	Berhasil

Berdasarkan tabel 2, dapat dikatakan bahwa hasil pengujian dengan skenario history payment telah berjalan dengan baik sesuai alternatif inputan yang telah dilakukan.

Berikut contoh bukti transfer bank yang telah berhasil dilakukan. Uji coba dilakukan dengan membuat data santri baru dengan nama PP Puncak Darussalam dengan nomor virtual account yang telah berhasil di buat yaitu 900518820000865. Kemudian dilakukan transfer uang melalui mobile banking dengan nominal Rp. 10.000,-



Gambar 13. Bukti Transfer Bank

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil pengembangan dan uji coba yang sudah dikerjakan pada aplikasi saldo yang telah di buat, dapat diambil kesimpulan bahwa penggunaan metode RAD (Rapid Application Development), menjadikan proses perancangan dan pembuatan aplikasi menjadi lebih cepat dan terstruktur, ini dikarenakan metode RAD harus melakukan secara terurut modul yang telah ditentukan sehingga dapat mengurangi kesalahan yang terjadi pada saat pembuatan modul dikarenakan harus menyelesaikan terlebih dahulu proses pengujian setiap modul sebelum melanjutkan pada modul berikutnya.

Aplikasi telah berhasil dibuat dengan berbasis website. Pengujian fungsionalitas aplikasi menggunakan metode black box testing telah digunakan dalam penelitian ini, dengan melakukan pengujian sesuai kebutuhan fitur aplikasi yang telah dirancang sebelumnya. Berdasar hasil pengujian, dapat dilihat bahwa aplikasi ini telah berfungsi dengan baik dan sudah sesuai dengan rancangan yang diinginkan, terlihat dari hasil pengujian, semua fitur yang diuji memiliki kesimpulan berhasil (dapat lihat pada tabel 1 dan tabel 2).

Aplikasi ini telah menjawab keinginan Pengasuh Pondok Pesantren serta memberikan solusi agar Wali Santri tidak perlu berkunjung ke Pondok Pesantren jika hanya ingin memberikan uang kepada anaknya.

5.2. Saran

Untuk membangun aplikasi seperti ini diperlukan banyak sumber daya seperti harus bekerja sama terlebih dahulu dengan pihak Bank, kemudian melakukan persamaan persepsi terhadap desain arsitektur yang akan dibuat, baru kemudian pihak Bank membuatkan API sesuai dengan kebutuhan, setelah itu menyesuaikan API yang telah dibuat untuk dikonversi menjadi modul modul aplikasi yang akan kita buat. Sehingga dibutuhkan kerjasama yang baik antara kedua pihak.

DAFTAR PUSTAKA

- [1] M. Danuri, "Perkembangan Dan Transformasi Teknologi Digital," *Infokam*, vol. XV, no. II, pp. 116–123, 2019.
- [2] A. W. Syahroni and Ubaidi, "Perancangan Aplikasi E-Money Dan Sms Gateway Untuk Pondok Pesantren Di Daerah Madura," *J. Link*, vol. 27, no. 1, pp. 1–7, 2018.
- [3] Y. Efendi, "Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile," *J. Ilm. Ilmu Komput.*, vol. 4, no. 1, pp. 19–26, 2018.
- [4] P. Bella Ramadhanu and A. T. Priandika, "Rancang Bangun Web Service API Aplikasi Sentralisasi Produk UMKM Pada UPTD PLUT KUMKM Provinsi Lampung," *J. Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 59–64, 2021.
- [5] R. Choirudin and A. Adil, "Implementasi Rest API Web Service Dalam Membangun Aplikasi Multiplatform Untuk Usaha Jasa," *J. MATRIK*, vol. 18, no. 2, pp. 284–293, 2019.
- [6] D. Setiadi and M. Nurdin Abdul Muhaemin, "Penerapan Internet Of Things (IoT) Pada Sistem Monitoring Irigasi (Smart Irigasi)," *J. Infotronik*, vol. 3, no. 2, pp. 95–102, 2018.

- [7] U. Rahardja, Q. Aini, N. Puji, and L. Santoso, "Peningtegrasian YII Framework Berbasis API pada Sistem Penilaian Absensi Integration of API-Based YII Framework on Attendance Assessment System," *J. Ilm. SISFOTENIKA*, vol. 8, no. 2, pp. 140–152, 2018.
- [8] R. Parluka and A. Pratama, "Aplikasi Penampil Berita Realtime Berbasis Bot Telegram Menggunakan API Web (APBR Versi 1.0)," *Semin. SANTIKA*, no. ISBN : 978-6-02-526748-2, 4-5 SEPTEMBER 2019, pp. 17–20, 2019.
- [9] M. E. Indrawan and A. Adil, "Implementasi Restful Web Service One Chip Multi-Client Untuk Mengoptimalkan Penjualan Pulsa All Operator," *J. Matrik*, vol. 16, no. 2, pp. 11–16, 2017.
- [10] W. R. F. Muhammad and S. F. S. Gumilang, "Rancangan Web Service Dengan Metode REST API Untuk Integrasi Aplikasi Mobile dan Website Pada Bank Sampah," *Konf. Nas. Sist. Inf. 2018 STMIK Atma Luhur Pangkalpinang*, no. 8-9 Maret 2018, pp. 214–219, 2018.
- [11] L. Efrizoni, R. K. Soemanov, and Y. Efendi, "Virtual Account dan Tracer Digital Untuk Pembayaran Uang Sekolah Siswa Berbasis Android," *Semin. Nas. APTIKOM*, pp. 80–90, 2019.
- [12] R. Sutjiadi, T. Rahmawati, and E. Halim, "Pengembangan Website Marketplace Binatang Peliharaan dengan Fitur Lelang Menggunakan Metode Rapid Application Development," *J. Sist. Inf. Bisnis*, vol. 02, no. 2021, pp. 152–160, 2021.
- [13] A. W. Syahroni and N. Ramadhani, "Perancangan Aplikasi Manajemen Penjualan Pada Apotek," *ANTIVIRUS J. Ilm. Tek. Inform.*, vol. 15, no. 1, pp. 73–83, 2021.