

Black Box Testing Aplikasi Pelayanan Permintaan dan Pengiriman Material PT Bank Rakyat Indonesia (Persero) Tbk

Mochamad Nandi Susila¹, Muhammad Darussalam²

Abstract—PT.Bank Rakyat Indonesia (Persero) Tbk has a work unit spread all over Indonesia. The Retail Business Network Division is one of the divisions within it, responsible for the demand and delivery of materials to existing work units. Application of demand and delivery of check material applications, bilyet giro and deposits, BRI's work environment is required to have system reliability in terms of application functionality. In this research, application development uses diagram design in UML (Unified Modeling Language) as an application architecture design and CodeIgniter framework for program code editor. While application testing is done by blackbox testing techniques with equivalent partitioning techniques, boundary value analysis, cause effect graphic. The application is able to produce service request report and delivery of the material as a whole, and facilitate employees in getting information about material management checks, bilyet giro and deposit, because the reliability of the application function has been supported by testing black box testing.

Intisari—PT.Bank Rakyat Indonesia (Persero) Tbk memiliki unit kerja yang tersebar diseluruh wilayah Indonesia. Divisi Jaringan Kerja Bisnis Ritel merupakan salah satu divisi didalamnya, yang bertanggung jawab terhadap kegiatan permintaan dan pengiriman material kepada unit-unit kerja yang ada. Penerapan aplikasi permintaan dan pengiriman material cek, bilyet giro dan deposito, dilingkungan kerja BRI diharuskan memiliki kehandalan sistem dari sisi fungsionalitas aplikasi. Dalam penelitian ini, Pengembangan aplikasi menggunakan rancangan diagram-diagram pada UML (Unified Modelling Language) sebagai desain arsitektur aplikasi serta framework CodeIgniter untuk editor kode program. Sementara Pengujian aplikasi dilakukan dengan teknik blackbox testing dengan teknik equivalence partitioning, boundary value analysis, cause effect graphic. Aplikasi yang diterapkan mampu menghasilkan laporan pelayanan permintaan dan pengiriman material secara keseluruhan, serta mempermudah karyawan dalam mendapatkan informasi tentang pengelolaan material cek, bilyet giro dan deposito, karena kehandalan fungsionalitas aplikasi telah didukung oleh pengujian black box testing.

Kata Kunci— PT.Bank Rakyat Indonesia, Aplikasi Permintaan dan Pengiriman Material, Black Box Testing

¹Program Studi Teknik Informatika, STMIK Antar Bangsa, Kawasan Bisnis CBD Ciledug. Jl.HOS Cokroaminoto Blok A5 No.29-36 Karang Tengah. Tangerang 15151 (telp: 021-7345-3000; e-mail : mmandisusila@gmail.com)

²Program Studi Manajemen Informatika, AMIK BSI Pontianak, Jl. Abdurahman Saleh no.A-18, Pontianak 78112; e-mail : muhammad.mds@bsi.ac.id

I. PENDAHULUAN

A. Latar Belakang Penelitian

Perbankan merupakan badan usaha yang menghimpun dana dari masyarakat dalam bentuk simpanan dan menyalurkannya kepada masyarakat dalam bentuk kredit dan atau bentuk-bentuk lainnya dalam rangka meningkatkan taraf hidup rakyat banyak. PT. Bank Rakyat Indonesia memiliki Kantor Pusat di Jakarta sedangkan unit kerja operasionalnya tersebar di seluruh Indonesia dari Sabang sampai Merauke. Divisi Jaringan Kerja Bisnis Ritel adalah divisi *support* yang bertempat di Kantor Pusat BRI Jakarta. Tugas utama dari Divisi ini adalah melakukan pembukaan unit kerja ritel sesuai dengan rencana bisnis bank yang diterbitkan oleh Bank Indonesia. Divisi Jaringan Kerja Bisnis Ritel dibantu oleh Kantor Wilayah yang tersebar di 19 Kantor Wilayah BRI diseluruh Indonesia.

Divisi Jaringan Kerja Bisnis Ritel dalam proses kerjanya masih dengan surat menyurat, seperti permintaan material Cek, Bilyet Giro, dan Deposito serta pendistribusian kebutuhan unit kerjadan lain-lain. Proses kerja yang dilakukan tersebut tentunya mempunyai banyak kendala-kendala yang dihadapi dalam Divisi Jaringan Kerja Bisnis Ritel, yaitu dalam hal permintaan dan pendistribusian material Cek, Bilyet Giro, dan Deposito dikarenakan harus melewati beberapa tahapan yang harus dilakukan oleh unit kerja, diantaranya unit kerja harus menyurati kantor wilayah terlebih dahulu kemudian diteruskan ke kantor pusat agar dapat dipenuhi kebutuhannya.

Adanya penerapan aplikasi permintaan dan pengiriman material cek, bilyet giro dan deposito, dilingkungan kerja BRI diharuskan memiliki kehandalan sistem dari sisi fungsionalitas aplikasi. Berdasarkan hal tersebut penerapan aplikasi membutuhkan adanya metode pengujian aplikasi agar kebutuhan fungsionalitas mampu berjalan dengan baik.

B. Rumusan Masalah

Melihat dari latar belakang penelitian diatas, beberapa rumusan masalah yang peneliti dapat adalah:

1. Bagaimana merancang dan menerapkan aplikasi yang dapat melayani permintaan dan pengiriman material seperti cek, bilyet giro dan deposito serta menyajikan laporan baik untuk Kantor Cabang dan Kantor Wilayah PT. Bank Rakyat Indonesia (Persero) Tbk?

2. Bagaimana metode pengujian aplikasi yang tepat mampu menganalisa kehandalan sistem dari kebutuhan fungsionalitas yang ada?

C. Batasan Masalah

Berdasarkan rumusan-rumusan masalah diatas, maka peneliti memiliki batasan masalah dalam penelitian ini, yaitu hanya pada penerapan aplikasi permintaan dan pengiriman material yang dilakukan pada Divisi Jaringan Kerja Bisnis Ritel saja. Kemudian melakukan pengujian aplikasi dengan metode pengujian *black box testing*.

D. Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini yaitu menerapkan dan melakukan pengujian pada aplikasi pelayanan permintaan dan pengiriman material pada PT. Bank Rakyat Indonesia (Persero) Tbk. Sedangkan manfaat dari penelitian ini adalah monitoring pendistribusian material seperti cek, bilyet giro maupun deposito dapat dilakukan dengan operasional yang lebih baik dan terintegrasi secara teknologi karena kehandalan aplikasi yang diterapkan.

II. KAJIAN LITERATUR

A. Pengelolaan Aset

Aset bagi sebuah bank adalah kekayaan bank berupa penempatan dana dalam bentuk kas atau uang tunai, giro pada bank Indonesia, giro pada bank lain surat berharga, pinjaman dari pemegang saham dan lembaga lainnya [4].

Akses ke dalam sistem komputer atau menjalankan operasional dengan baik, setiap bank mempunyai sistem aplikasi utama (*core banking system*) masing-masing sesuai dengan kebutuhan dan bisnis prosesnya. Komponen distribusi informasi, khususnya didunia perbankan terdiri dari:

1. *Offline*, dimana distribusi berdiri sendiri tidak mempunyai hubungan antara satu dengan bagian lainnya
2. *Online*, dimana distribusi mempunyai keterkaitan hubungan antara satu dengan bagian lainnya. Saluran dengan memanfaatkan saluran intranet atau internet. Akses tersebut terbatas, biasanya diatur dalam ketentuan dan prosedur mengenai kode unik untuk mengidentifikasi (*user-id*) dan sistem pengaman berupa sandi rahasia (*password*) [4].

B. UML (*Unified Modelling Language*)

UML singkatan dari *Unified Modeling Language*, UML adalah kosakata umum berbasis objek dan diagram teknik yang cukup efektif untuk memodelkan setiap proyek pengembangan sistem mulai tahap analisis sampai tahap perancangan dan tahap implementasi[2]. Berikut penjelasan beberapa diagram UML yang dilakukan dalam penelitian ini:

1. Diagram *Use Case*

Diagram *Use Case* merupakan suatu diagram yang menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi antara sistem dan lingkungannya[2].

TABEL 1.
SIMBOL DIAGRAM *USE CASE*

Term and Definition	Symbol
<p>An actor</p> <ul style="list-style-type: none"> Is a person or system that derives benefit from and is external to the system. Is labeled with its role. Can be associated with other actors by a specialization/superclass association, denoted by an arrow with a hollow arrowhead. Is placed outside the system boundary. 	
<p>A use case</p> <ul style="list-style-type: none"> Represents a major piece of system functionality. Can extend another use case. Can use another use case. Is placed inside the system boundary. Is labeled with a descriptive verb-noun phrase. 	
<p>A system boundary</p> <ul style="list-style-type: none"> Includes the name of the system inside or on top. Represents the scope of the system. 	
<p>An association relationship</p> <ul style="list-style-type: none"> Links an actor with the use case(s) with which it interacts. 	

- Aktor (*An actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- A *Use case*, aktifitas / sarana yang disiapkan oleh bisnis / sistem.
- A *System boundary*, sebuah kotak yang mewakili sebuah sistem.
- Hubungan (*An association relationship*), aktor mana saja yang terlibat dalam *use case*, dan bagaimana hubungan *use case* dengan *use case* lain. ada hubungan antar *use case*. Digolongkan menjadi 2 : yaitu *extend* digambarkan dengan keterangan <<extend>>, dan *include* digambarkan dengan keterangan <<include>>

2. Diagram Aktifitas

Diagram aktifitas merupakan diagram yang menggambarkan alur kerja bisnis independen dari kelas, aliran kegiatan dalam *use case* atau desain rinci sebuah metode[2].

3. Diagram *Sequence*

Sequence Diagram merupakan urutan model dinamis yang menggambarkan contoh kelas yang berpartisipasi dalam *use case* dan pesan yang lewat di antara mereka dari waktu ke waktu[2]

4. Diagram *Class*

Diagram kelas atau *Class* Diagram adalah model statis yang mendukung pandangan statis dari sistem berkembang. Ini menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. Diagram kelas sangat mirip dengan diagram hubungan entitas (ERD) [2].

C. *Framework CodeIgniter*

Framework CodeIgniter adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun website dinamis dengan menggunakan PHP[7]. *CodeIgniter* memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.

Model View *Controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu :

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu *programmer* yang menangani bagian model dan *controller*, sedangkan *designer* yang menangani bagian *view*, sehingga penggunaan arsitektur MVC dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara *programmer* dan *designer* dalam menangani variabel-variabel yang akan ditampilkan.

Ada beberapa kelebihan CodeIgniter (CI) dibandingkan dengan *Framework PHP* lain,

1. Performa sangat cepat : salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP, tapi Codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan *framework* yang paling cepat dibanding *framework* yang lain.
2. Konfigurasi yang sangat minim (*nearly zero configuration*) : tentu saja untuk menyesuaikan dengan database dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti database.php atau autoload.php, namun untuk menggunakan codeigniter dengan pengaturan standar, anda hanya perlu mengubah sedikit saja file pada folder config.
3. Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
4. Dokumentasi yang sangat lengkap : Setiap paket instalasi codeigniter sudah disertai user guide yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

D. MySQL

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Sehingga istilah seperti tabel, baris, dan kolom tetap digunakan dalam MySQL[1]. MySQL sebuah *database* mengandung satu beberapa tabel, tabel terdiri dari sejumlah baris dan kolom. Sedangkan *database* sendiri dapat didefinisikan sebagai kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan[3].

Mysql dikembangkan oleh sebuah perusahaan Swedia bernama mysql AB.. dapat dilihat dan gratis, serta servermysql dapat dipakai tanpa biaya tapi hanya untuk kebutuhan nonkomersial.

E. Black Box Testing

Dalam penelitian ini, peneliti menggunakan metode pengujian *blackbox*. *Blackbox Testing* adalah tipe pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses testing di bagian luar”[5].

Beberapa alasan keuntungan digunakannya metode pengujian ini oleh peneliti, yaitu:

1. Penguji tidak harus dari seseorang yang memiliki kemampuan teknis di bidang pemrograman.
2. Kesalahan dari perangkat lunak ataupun bug seringkali ditemukan oleh komponen tester yang berasal dari pengguna.
3. Hasil dari *blackbox* testing dapat memperjelas kontradiksi ataupun keracunan yang mungkin timbul dari eksekusi sebuah perangkat lunak.
4. Proses *testing* dapat dilakukan lebih cepat dibandingkan *whitebox testing*.

Pengujian *blackbox Texting* lebih focus kepada hasil yang diinginkan tanpa harus memikirkan proses internal atau bahkan *source code* yang ada didalam aplikasi tersebut. Pengujian dari hasil dilakukan melalui data uji dan pemeriksaan fungsional dari aplikasi itu sendiri.

Secara umum, pengujian *black box* itu menguji aspek fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerjanya.

Beberapa teknik yang biasa digunakan untuk merancang pengujian *black box*:

1. *Equivalence Partitioning*
Dalam teknik ini, pengujian akan melibatkan pembagian nilai input kedalam bagian nilai valid dan tidak valid serta pemilihan data *test*.
2. *Boundary Value Analisis*
Dalam teknik ini, pengujian akan melibatkan penentuan-penentuan nilai input dan memilih beberapa nilai dari batasan-batasan tersebut baik luar maupun dalam dari batasan-batasan tersebut sebagai data *test*.
3. *Cause Effect Graphic*
Dalam teknik ini, pengujian akan melibatkan identifikasi kondisi input (percabangan ataupun perulangan)

III. METODE PENELITIAN

Berikut beberapa tahapan dalam pengembangan dan pengujian aplikasi yang peneliti lakukan dalam penelitian ini:

A. Tahapan Pengembangan Aplikasi

Sebelum melakukan pengembangan aplikasi, peneliti mengumpulkan data-data yang dibutuhkan terlebih dahulu dengan cara studi *literature* dan observasi kepada *stakeholder*. Dari data-data yang telah dikumpulkan, peneliti melakukan analisa kebutuhan aplikasi yang diperlukan, seperti kebutuhan-kebutuhan fungsional maupun kebutuhan-kebutuhan non fungsional.

Sementara untuk tahapan dokumentasi aplikasi, peneliti menggunakan diagram UML, seperti diagram *use case*, diagram aktifitas, diagram *sequence*, hingga diagram *class* yang dijadikan acuan hubungan antar kelas dan objek untuk kemudian diterapkan kedalam bentuk spesifikasi tabel pada basis data yang dibutuhkan. Kemudian penerapan kode program menggunakan *framework* CodeIgniter.

B. Tahapan Pengujian *Black Box Testing*

Beberapa metode pengujian *black box* yang dilakukan oleh peneliti dalam penelitian ini, yaitu:

1. *Equivalence Partitioning*

Pada metode ini, peneliti memilih komponen-komponen dari nilai masukan (input) pada aplikasi. Nilai masukan akan diuji dengan perbandingan nilai yang valid maupun yang tidak valid, apakah sesuai dengan kebutuhan fungsional dari aplikasi.

2. *Boundary Value Analysis*

Pada metode ini, peneliti melanjutkan dari tahapan *equivalence partitioning* data *test* yang diujikan, namun dengan membandingkan batasan-batasan maksimum dan minimum dari input nilai data *test* yang diujikan.

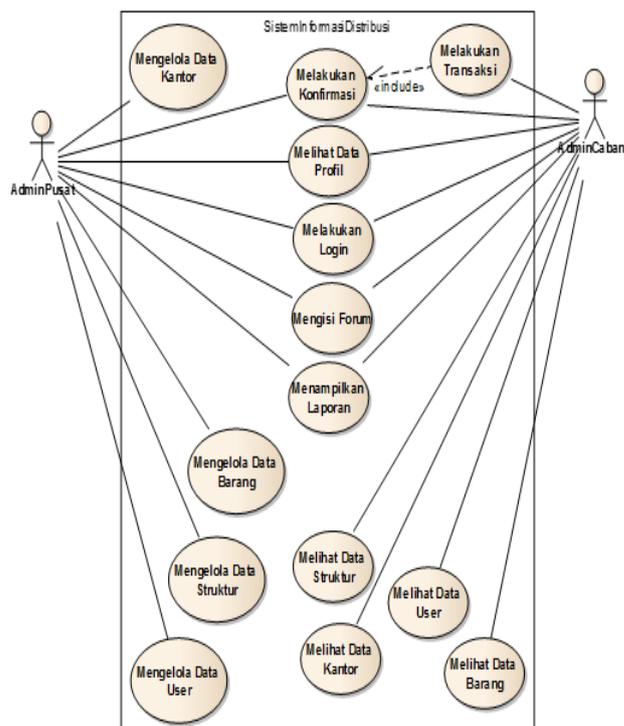
3. *Cause Effect Graphic*

Pada metode ini, peneliti memilih komponen-komponen dari data *test* dengan melakukan uji coba input pada kondisi yang berbeda, baik itu dalam kondisi percabangan maupun perulangan. Sehingga efek dari nilai input data *test* akan bisa dilihat dari grafis tampilan.

- c. Aplikasi dapat digunakan untuk mengelola data barang material
 - d. Aplikasi dapat digunakan untuk mengelola permintaan dan pengiriman material
 - e. Aplikasi dapat digunakan untuk pengelolaan laporan permintaan dan pengiriman material
2. Kebutuhan Non Fungsional
- a. Aplikasi dapat dijalankan pada beberapa *web browser* seperti contoh Internet Explore, Google Chrome ataupun Mozilla Firefox
 - b. Aplikasi harus dipastikan memiliki validasi dalam penginputan data
 - c. Aplikasi harus dipastikan memiliki hak akses *user* yang berwenang untuk pengelolaan data
 - d. Aplikasi harus memiliki *interface* (tampilan antar muka) yang mudah difahami dan digunakan (*usability*)

B. Diagram *Use Case*

Dibawah ini adalah hasil gambaran diagram *use case* dari penelitian yang dilakukan :



Gbr 1. Diagram *Use Case* Aplikasi

IV. HASIL DAN PEMBAHASAN

A. Analisa Kebutuhan Aplikasi

Tahapan analisa kebutuhan merupakan salah satu langkah penting dalam siklus hidup pengembangan suatu aplikasi. Didalam tahapan ini memastikan aplikasi yang akan dirancang bangun sesuai dengan kebutuhan dari sisi pengguna.

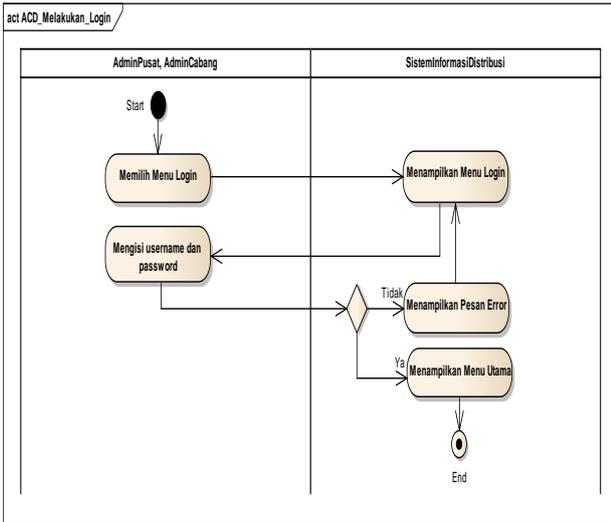
Berikut ini beberapa hasil analisa kebutuhan aplikasi yang dibutuhkan pada penelitian ini:

1. Kebutuhan Fungsional

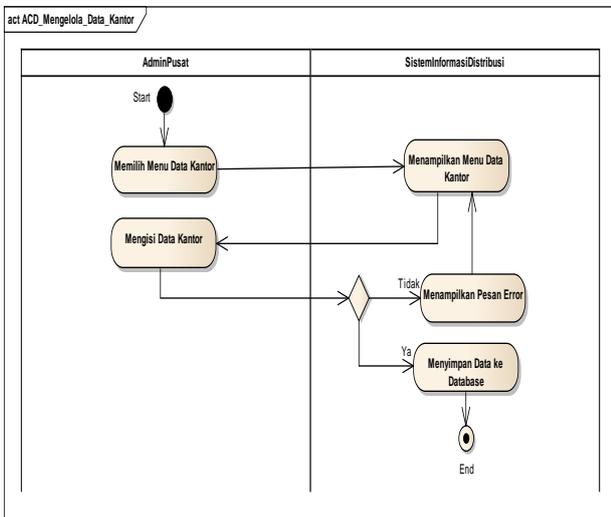
- a. Aplikasi dapat digunakan untuk mengelola data kantor
- b. Aplikasi dapat digunakan untuk mengelola diskusi forum

C. Diagram Aktifitas

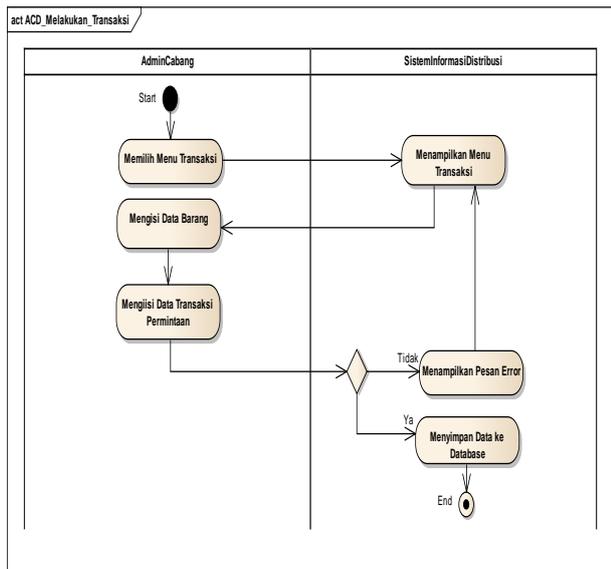
Dibawah ini adalah hasil gambaran diagram aktifitas dari penelitian yang dilakukan :



Gbr 2. Diagram Aktifitas Login Admin



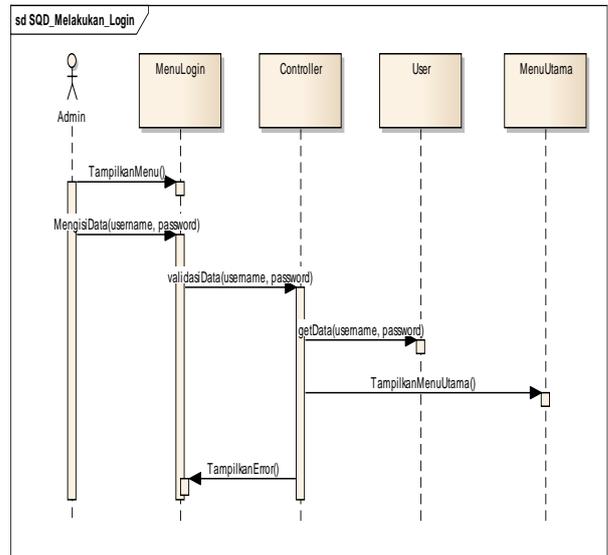
Gbr 3. Diagram Aktifitas Pengelolaan Data Kantor



Gbr 4. Diagram Aktifitas Permintaan Material

D. Diagram Sequence

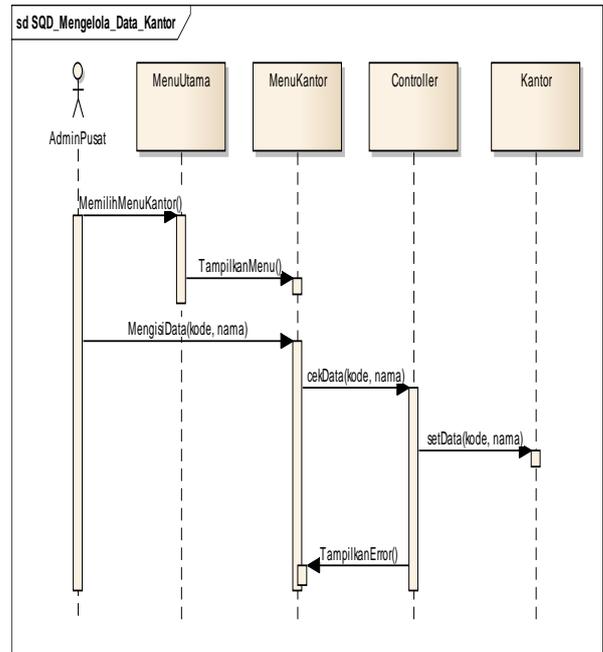
Dibawah ini adalah hasil gambaran diagram *sequence* dari penelitian yang dilakukan :



Gbr 5. Diagram Sequence Login Admin

TABEL 2.
SKENARIO LOGIN ADMIN

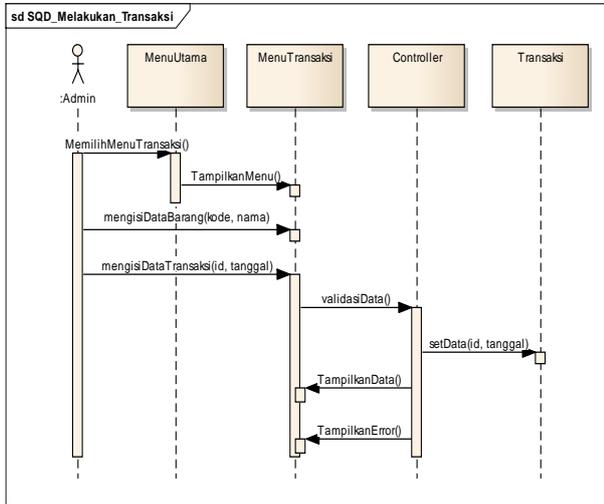
Aktor	Admin Pusat, Admin Cabang
Skenario	Memasukkan <i>Username</i> dan <i>password</i> kemudian klik tombol login. Jika <i>username</i> atau <i>password</i> salah, akan menampilkan pesan error, dan jika <i>username</i> dan <i>password</i> benar akan menampilkan halaman utama sesuai level masing-masing admin.



Gbr 6. Diagram Sequence Pengelolaan Data Kantor

TABEL 3.
SKENARIO PENGELOLAAN DATA KANTOR

Aktor	Admin Pusat
Skenario	Memilih menu Data Kantor, mengisi data kantor dengan lengkap, kemudian menyimpan data ke <i>database</i>



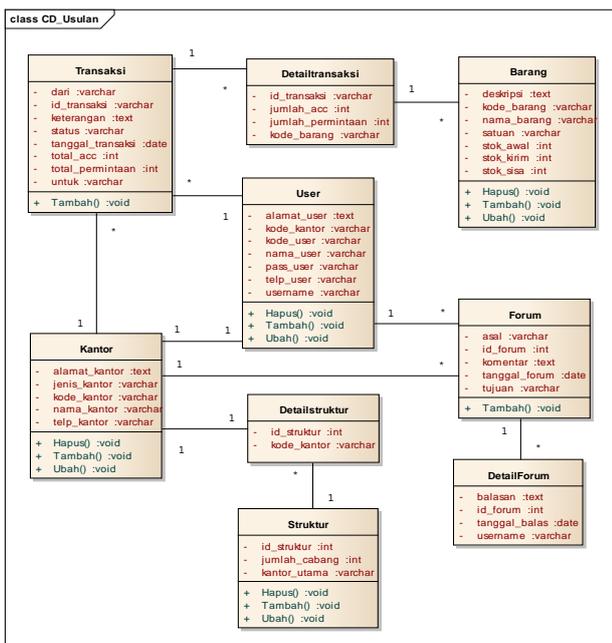
Gbr 7. Diagram Sequence Permintaan Material

TABEL 4.
SKENARIO PENGELOLAAN DATA KANTOR

Aktor	Admin Wilayah, Admin Cabang
Skenario	Memilih menu Transaksi Permintaan, mengisi data transaksi permintaan dengan lengkap, kemudian menyimpan data ke database

E. Diagram Class

Dibawah ini adalah hasil gambaran diagram *class* dari penelitian yang dilakukan:



Gbr 8. Diagram Class Aplikasi

F. Tampilan Aplikasi

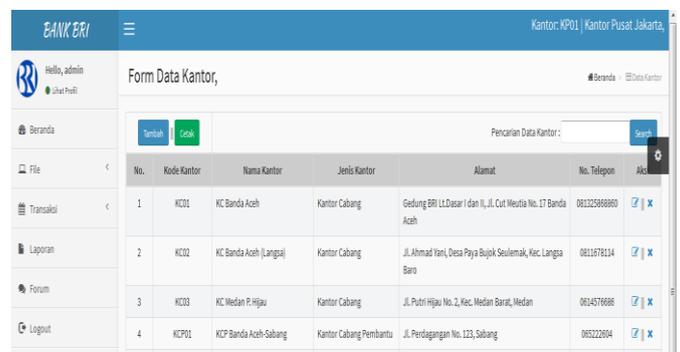
Berikut beberapa tampilan dari aplikasi hasil penelitian yang peneliti buat:

1. Tampilan Halaman Login



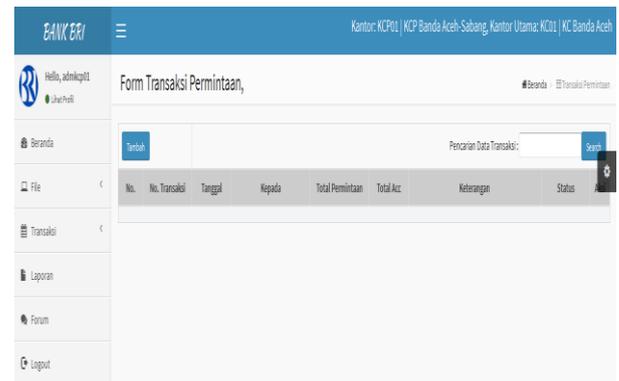
Gbr 9. Tampilan Login User

2. Tampilan Halaman Pengelolaan Data Kantor



Gbr 10. Tampilan Pengelolaan Data Kantor

3. Tampilan Halaman Permintaan Material

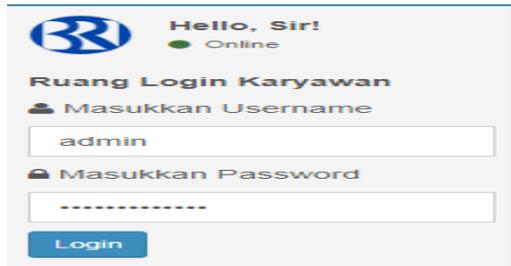


Gbr 11. Tampilan Permintaan Material

G. Pengujian Black Box Testing

Berikut hasil dari pengujian *black box testing* terhadap aplikasi dalam penelitian ini:

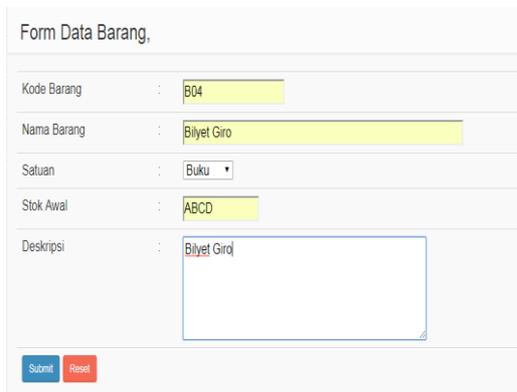
1. Pengujian *Equivalence Partitioning*



Gbr 12. Uji *Equivalence Partitioning* Login User

TABEL 5.
SKENARIO DAN HASIL PENGUJIAN *EQUIVALENCE PARTITIONING* LOGIN USER

No	Halaman	Test Case	Hasil yang diharapkan	Skenario Pengujian	Hasil
1	Halaman login user	Menguji input data test username dan password, dengan karakter huruf dan angka sesuai dengan data yang ada pada basis data	Berhasil masuk kedalam halaman utama aplikasi user	Memasukan username dan password dengan huruf maupun angka yang sesuai pada basis data	Inputan valid baik berupa huruf atau angka untuk login user
2	Halaman login user	Menguji input data test username dan password, dengan karakter huruf dan angka tidak sesuai dengan data yang ada pada basis data	Aplikasi menahan hasil input dan menolak untuk masuk kedalam halaman utama aplikasi user	Memasukan username dan password dengan huruf maupun angka yang sesuai dengan data pada basis data	Inputan valid baik berupa huruf atau angka untuk login user



Gbr 13. Uji *Equivalence Partitioning* Form Data Barang

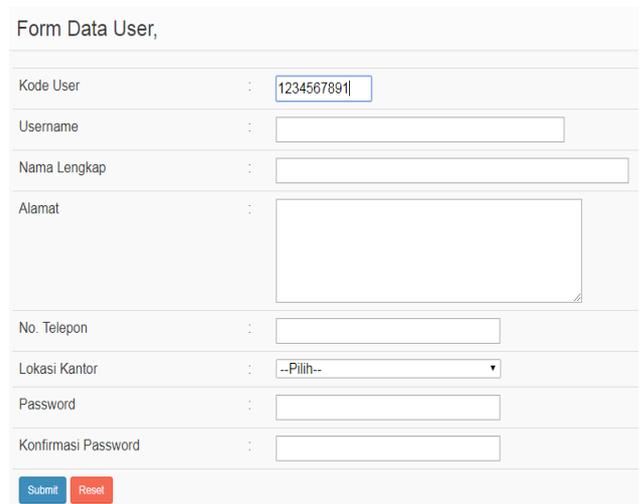


Gbr 14. Hasil Validasi Uji *Equivalence Partitioning* Form Data Barang

TABEL 6.
SKENARIO DAN HASIL PENGUJIAN *EQUIVALENCE PARTITIONING* FORM DATA BARANG

No	Halaman	Test Case	Hasil yang diharapkan	Skenario Pengujian	Hasil
1	Halaman Form Data Barang	Menguji input data test Stok Awal dengan karakter huruf	Akan ada pengecekan validasi input data test, dan untuk tidak masuk kedalam sistem berikutnya	Memasukan input dengan huruf atau gabungan huruf	Uji data test akan ada validasi input karena tidak sesuai dengan kebutuhan fungsional aplikasi
2	Halaman Form Data Barang	Menguji input data test Stok Awal dengan karakter angka	Akan ada pengecekan validasi input data test, dan berhasil masuk kedalam sistem berikutnya	Memasukan input Stok Awal dengan angka	Uji data test akan ada validasi input dan sesuai dengan kebutuhan fungsional aplikasi

2. Pengujian *Boundary Value Analysis*



Gbr 15. Uji *Boundary Value Analysis* Form Data User

TABEL 7.
SKENARIO DAN HASIL PENGUJIAN *BOUNDARY VALUE ANALYSIS*
FORM DATA *USER*

No	Halaman	Test Case	Hasil yang diharapkan	Skenario Pengujian	Hasil
1	Halaman Form Data <i>User</i>	Menguji input data <i>test Kode User</i> dengan melebihi batasan maksimal <i>size</i> 10 karakter sesuai kebutuhan fungsional	Aplikasi akan menahan untuk tidak dapat melanjutkan ketahapan proses selanjutnya	Memasukkan input Kode <i>User</i> lebih dari 10 karakter	Aplikasi akan menahan dan input data hanya bisa maksimal 10 karakter

Gbr 16. Uji *Boundary Value Analisis* Login Admin

TABEL 8.
SKENARIO DAN HASIL PENGUJIAN
BOUNDARY VALUE ANALYSIS LOGIN ADMIN

No	Halaman	Test Case	Hasil yang diharapkan	Skenario Pengujian	Hasil
1	Halaman Form Data Barang	Menguji input data <i>test</i> Stok Awal dengan melebihi batasan maksimal <i>size</i> 4 karakter sesuai kebutuhan fungsional	Aplikasi akan menahan untuk tidak dapat melanjutkan ketahapan proses selanjutnya	Memasukkan input Stok Awal lebih dari 4 karakter	Aplikasi akan menahan dan input data hanya bisa maksimal 4 karakter

3. Pengujian *Cause Effect Graphic*

Gbr 17. Uji *Cause Effect Graphic* Halaman Utama Admin

TABEL 9.
Skenario dan Hasil Pengujian *Cause Effect Graphic* Halaman Utama Admin

No	Halaman	Test Case	Hasil yang diharapkan	Skenario Pengujian	Hasil
1	Halaman Login Admin	Menguji akses masuk kedalam halaman utama admin, dengan pembacaan percabangan data yang dibutuhkan yang tidak dibutuhkan	Berhasil masuk kedalam Halaman Utama Admin	Pengujian dengan melakukan input data <i>test</i> jika sesuai dengan kebutuhan aplikasi	Hasil dari pengecekan kondisi pilihan sesuai dan tidak sesuai akan menghasilkan tampilan grafis yang diharapkan

Gbr 18. Uji *Cause Effect Graphic* Login Admin

Gbr 19. Uji *Cause Effect Graphic* Login Admin

TABEL 10.
SKENARIO DAN HASIL PENGUJIAN *CAUSE EFFECT GRAPHIC*
LOGIN ADMIN

No	Halaman	Test Case	Hasil yang diharapkan	Skenario Pengujian	Hasil
1	Halaman Form Data Barang	Menguji tombol-tombol yang terdapat dalam antar muka tersebut, dan kesesuaian hak akses dengan login user	Laman login sudah terbuka	a. Memasukkan user dan password yang benar b. Memasukkan user dan password yang salah	Data akan dicek user akan berhasil masuk ke aplikasi user dan password valid

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, maka peneliti dapat menarik kesimpulan:

1. Aplikasi yang diterapkan dapat menghasilkan laporan pendistribusian material, baik permintaan maupun pengiriman secara keseluruhan, serta mempermudah karyawan dalam mendapatkan informasi tentang pengelolaan material cek, bilyet giro dan deposito.
2. Pengujian *black box testing* mampu menganalisa uji kebutuhan fungsionalitas dari aplikasi yang diterapkan.
3. Perlunya pengembangan kedepan untuk aplikasi agar dapat berjalan pada platform khusus *mobile*.

REFERENSI

- [1] Arief, M.Rudianto. 2011. Pemrograman Web Dinamis Menggunakan Php dan Mysql. Yogyakarta: Andi
- [2] Dennis, A., Wixom, B. H., & Roth, R. M. 2012. System Analysis and Design. United States of America: John Wiley & Sons, Inc.
- [3] Fathansyah. 2012. Basis Data. Bandung: Informatika.
- [4] Lembaga Sertifikat Profesi Perbankan (LSPP), Ikatan Bankir Indonesia (IBI), 2013. Memahami Bisnis Bank Ikatan Bank Indonesia Jakarta: PT Gramedia Pustaka Utama.
- [5] Rizky, Soetam. 2011. Konsep Dasar Rekayasa Perangkat Lunak. Jakarta: Prestasi Pustaka.
- [6] Rosa, A.S dan M. Shalahuddin. 2013. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung : Informatika.
- [7] Sidik, Betha Ir. 2012. Framework Code Igniter. Bandung: Informatika



Mochamad Nandi Susila, Lulus Pasca Sarjana Magister Ilmu Komputer Pada Tahun 2015 Konsentrasi Manajemen Sistem Informasi di STMIK Nusa Mandiri Jakarta. Saat ini aktif sebagai Dosen Tetap di STMIK Antar Bangsa.



Muhammad Darussalam, Lulus Pasca Sarjana Magister Ilmu Komputer Pada Tahun 2015 Konsentrasi Manajemen Sistem Informasi di STMIK Nusa Mandiri Jakarta.