Jurnal Ilmiah Komputer dan Informatika
Edisi...Volume..., Bulan 20..ISSN :2089-9033

AUTOMATIC DEPLOYMENT SYSTEM DENGAN MENGGUNAKAN METODE *CONTINUOUS INTEGRATION* DI KAKATU

Jaga My Priera[1], Robi Tanzil Ganefi[2]

[1, 2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : jagamyp@gmail.com[1], robi.tanzil.g@gmail.com[2]

ABSTRAK

PT. Kinest Kreatif Ideata atau yang lebih dikenal dengan Kakatu merupakan perusahaan yang bergerak di bidang Informasi Teknologi (IT). Produk dan layanan PT. Kinest Kreatif Ideata adalah sebuah aplikasi Android parental kontrol berupa *launcher* yang bernama Kakatu. Dengan menggunakan Kakatu, para orang tua dapat mengontrol pemakaian gadget oleh anak. Berdasarkan hasil wawancara dengan pihak Kakatu, diketahui bahwa proses pengembangan aplikasi Android di Kakatu dibagi menjadi beberapa tahapan yaitu analisis, *development*, *testing* dan *deployment*. Dari tahapan-tahapan tersebut ditemukan masalah pada proses development dan *testing*. Dimana pada tahap ini para *developer* menggunakan layanan pihak ketiga seperti Google Drive untuk mendistribusikan *file* apk ke tim penguji, dan Gmail sebagai media komunikasi antara tim penguji dan developer. Dengan tidak adanya sistem yang terpadu untuk melakukan kedua kegiatan tersebut menyebabkan susahnya proses pendokumentasian aplikasi yang dikembangkan.

Untuk mengatasi masalah tersebut maka dibangunlah Website Automatic Deployment dengan menggunakan Metode *Continuous Integration*(CI). CI merupakan sebuah metode dalam pengembangan perangkat lunak dimana tiap-tiap developer mengintegrasikan kode-kode aplikasi yang telah mereka buat kedalam sebuah repositori. Kode-kode tersebut kemudian dikompilasi dan dites. Kegiatan tersebut dilakukan secara otomatis setiap kali *developer* melakukan *commit* ke repositori. Salah satu perangkat lunak CI yang populer adalah Jenkins. Dengan banyaknya *plugin* yang dimiliki Jenkins, memungkinkan dibangunnya sistem CI yang kaya akan fitur. Salah satu *plugin* Jenkins yang sering digunakan adalah *plugin* untuk mengirimkan *file* executable via email secara otomatis setelah proses build berhasil.

Berdasarkan analisis dan pengujian yang telah dilakukan dengan melibatkan pihak Kakatu, dapat ditarik kesimpulan bahwa sistem ini telah layak digunakan oleh pihak Kakatu untuk membantu mempercepat dan mempermudah proses pengembangan perangkat lunak Android di sana.

Kata kunci : *Continuous Integration*, Jenkins, Kakatu, *Automatic Deployment*

1. PENDAHULUAN

Pada bagian ini akan dipaparkan mengenai latar belakang masalah, maksud dan tujuan.

1.1. Latar Belakang Masalah

PT. Kinest Kreatif Ideata atau yang lebih dikenal dengan Kakatu merupakan perusahaan startup yang berasal dari Bandung. Kakatu didirikan pada tahun 2014. Saat ini Kakatu memiliki dua produk dan layanan yaitu Kakatu Launcher dan Hyjabs. Kakatu merupakan aplikasi parental kontrol berplatform Android berbentuk *launcher*. Dengan menggunakan aplikasi ini, orang tua dapat mengontrol pemakaian gadget oleh anak. Sedangkan Hyjabs merupakan aplikasi media sosial untuk berbagi tren hijab terkini. Melalui Hyjabs, pengguna dapat saling berbagi tips, tren, dan tutorial seputar penggunaan hijab.

Berdasarkan hasil wawancara langsung dengan *developer* Android di Kakatu didapatkan beberapa fakta mengenai proses pengembangan aplikasi Android hingga ke tahap rilis.Pertama tiap-tiap *developer* menulis kode di komputer mereka masing-masing, kemudian kode tersebut dikompilasi lalu diuji, kalau tidak ada masalah, semua perubahan kode akan disimpan (*commit*) ke Bitbucket (*push*). Setelah itu kode akan disatukan (*merge*) lalu dikompilasi kembali, kemudian *file* apk hasil kompilasi terebut akan dikirim ke tim penguji. Kalau ada masalah tim penguji akan melaporkannya ke *developer*, kalau tidak ada masalah maka aplikasi siap untuk dirilis. Perangkat lunak yang digunakan selama dalam pengembangan aplikasi tersebut adalah Android Studio sebagai IDE, Git sebagai VCS, Bitbucket sebagai layanan manajemen repositori, Gmail sebagai media komunikasi antar anggota tim, dan Google Drive atau Dropbox sebagai media untuk mendistribusikan *file* apk ke anggota tim yang akan melakukan uji coba aplikasi. Selama proses pengembangan aplikasi tersebut ditemukan beberapa masalah, yaitu untuk membagikan *file* apk ke tim penguji, *developer* harus mengunggah file tersebut ke Google Drive atau Dropbox terlebih dahulu. Setelah file apk diunduh dan diuji, maka tim penguji akan

memberikan *feedback* melalui *email.*

Masalah lainnya terjadi pada saat melakukan dokumentasi *file* apk. Pada paragraf sebelumnya telah dijelaskan bahwa apk yang akan diuji dibagikan melalui Google Drive dan Dropbox, sedangkan laporan hasil uji coba apk ada di Gmail. Perbedaan lokasi tersebut menyembabkan susahnya mendokumentasikan *file* apk beserta laporan hasil uji coba.

Berdasarkan paparan masalah di atas, maka solusi yang ditawarkan adalah dengan membangun sistem yang dapat melakukan proses pembangunan kode yang terintegrasi menggunakan Jenkins serta membangun *website* untuk mendokumentasi hasil dari proses tersebut dengan memanfaatkan API yang dimiliki Jenkins. Pemilihan Jenkins sebagai perangkat lunak CI adalah karena Jenkins memiliki banyak *plugin* dan mendukung semua jenis repositori, baik itu Bitbucket, Gitlab, Github, bahkan repositori yang memiliki CVS selain Git seperti Launchpad yang menggunakan Bazaar. Selain itu, Jenkins bisa dipasang di *server* sendiri, sehingga pengguna bisa membangun sistem CI yg sesuai kebutuhan mereka.

1.2. Maksud dan Tujuan

Maksud dari penelitian ini adalah membangun *Automatic Deployment System* Dengan Menggunakan Metode *Continuous Integration* di Kakatu. Sedangkan tujuan yang akan dicapai yaitu:

1. Memudahkan proses pendistribusian aplikasi ke tim penguji.
2. Mempermudah komunikasi antara tim penguji dengan *developer*
3. Mempermudah proses dokumentasi *file* apk.

1.3. Continuous Integration (CI)

CI adalah suatu metode dalam pengembangan perangkat lunak, dimana kode dari tiap-tiap developer disatukan dan di-*build* secara berkala untuk mendeteksi kesalahan sedini mungkin.

Secara umum, tahapan-tahapan dalam sebuah sistem CI adalah sebagai berikut (dipicu dengan perintah *push* atau *commit*):

1. Integrated
Merupakan tahapan dimana tiap-tiap *developer* mengintegrasikan kode-kode atau hasil kerja mereka.

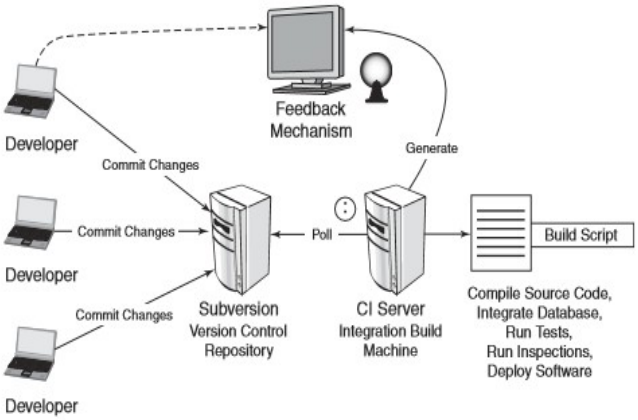
2. Compiled
Kode dikompilasi kedalam *paket* atau *executable*

3. Tested
Melakukan uji coba *executable* secara manual atau otomatis

4. Archived
Mengarsipkan *file excutable,* hasil tes, dan log selama proses tersebut berlangsung.

5. Deployed
Memasang atau membagikan hasil *build*.

Berikut gambar arsitektur sebuah sistem CI:



1.4. Jenkins

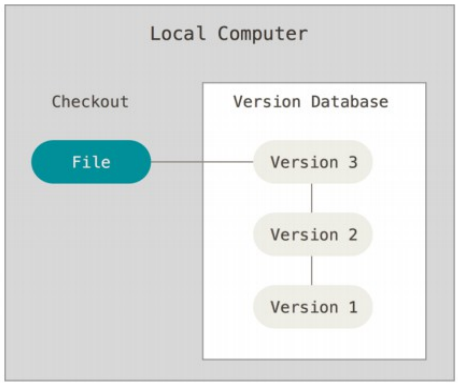
Jenkins merupakan salah satu *tool* CI yang dikembangkan oleh Kohsuke Kawaguchi dengan nama awal Hudson pada akhir 2004 ketika masih bekerja di Sun. Hudson berkembang selama beberapa tahun, sistem ini diadopsi oleh banyak tim dalam Sun untuk proyek-proyek mereka. Pada awal 2008, Sun mengakui kualitas dan nilai dari Hudson, dan meminta Kohsuke untuk bekerja pada Hudson penuh waktu, serta mulai memberikan layanan profesional dan mendukung Hudson. Pada tahun 2010, Hudson telah menjadi *tool* paling diminati dalam solusi *Continuous Integration* dengan pangsa pasar lebih dari 70%.

Pada bulan Januari 2011, komunitas pengembang Hudson memutuskan untuk mengubah nama proyek menjadi Jenkins. Mereka kemudian membuat salinan kode Hudson asli ke project GitHub baru dan meneruskan pekerjaan mereka di sana. Sebagian besar *developer* inti dan *plugin* beserta Kohsuke Kawaguchi dan kontributor inti bersama-sama mulai mengembangkan Jenkins.

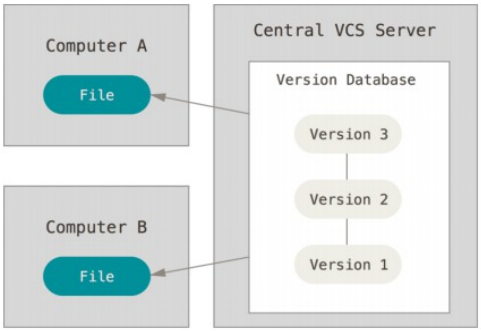
1.5. Version Control System (VCS)

Version Control Sistem adalah sebuah sistem yang mencatat perubahan file dari waktu ke waktu. Sebagai contoh, seorang web designer akan menyimpan setiap versi dari rancangan yang dibuatnya. Dengan menggunakan VCS memungkinkan untuk kembali ke berkas rancangan sebelumnya, membandingkan perubahan yang terjadi, serta melihat siapa yang telah membuat perubahan.

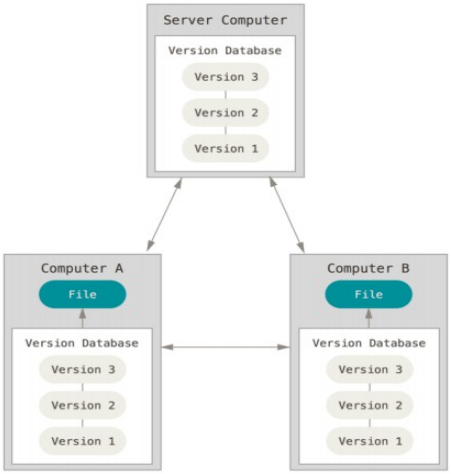
1. Local Version Control Systems (LCVS)



2. Centralized Version Control Systems (CVCS)



3. Distributed Version Control Systems (DVCS)



2. ISI PENELITIAN

2.1. Analisis dan Perancangan Sistem

Berikut adalah analisis perancangan dari sistem yang dibangun,

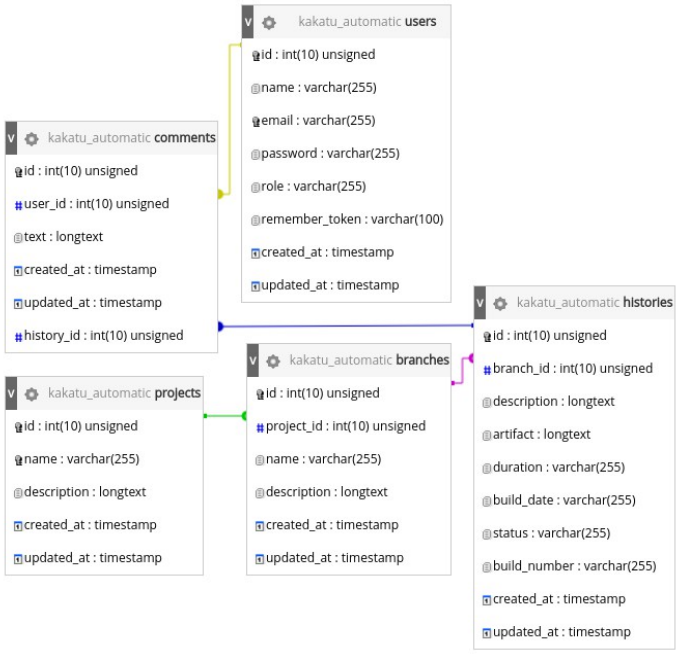
2.1.1. Analisis Masalah

Berikut rincian masalah yang terjadi pada sistem yang sedang berjalan:

1. Proses kompilasi kode dilakukan pada masing-masing komputer, hal ini menyebabkan masalah.
2. *File* apk yang dibagikan melalui Google drive atau Dropbox memerlukan langkah ekstra, seperti mengunggah *file* apk secara manual.
3. Pemberitahuan atau catatan mengenai masalah yang terjadi pada aplikasi yang dilakukan lewat *email* (Gmail) memerlukan langkah ekstra, seperti melampirkan versi *file* apk yang bermasalah.
4. Proses dokumentasi apk susah dilakukan. Karena *file* apk dan catatan mengenai *file* apk tersebut berada di lokasi yang berbeda.

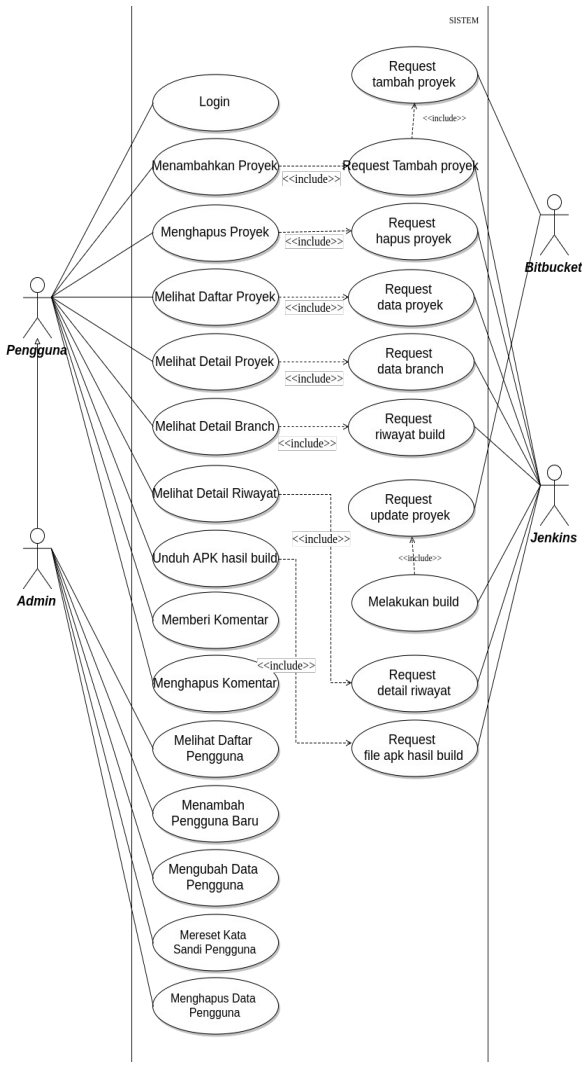
2.1.2. Analisis Data

Berikut adalah skema relasi sistem:

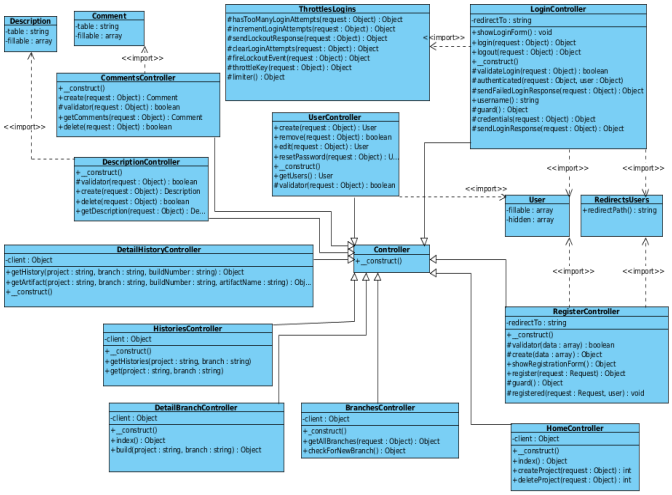


2.1.3. Analisis Kebutuhan Fungsionalitas

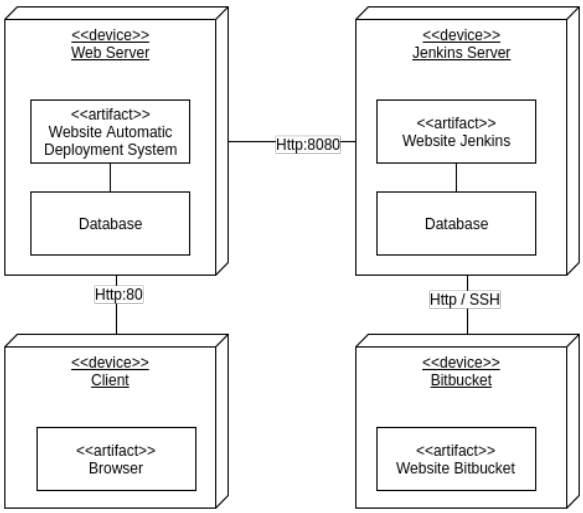
Berikut adalah diagram *use case* sistem:



Berikut adalah diagram kelas sistem:

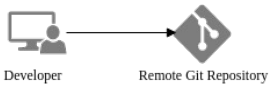


Berikut adalah diagram deployment sistem:



2.1.4. Analisis Proses *Build*

Jenkins akan memeriksa setiap *commit* terbaru yang ada di *remote* repositori secara berkala. Namun proses *build* hanya akan dieksekusi jika terdapat kata kunci "[ci build]" pada pesan *commit.*

Perintah Git	Aktor	Diagram
git commit -m "[ci *build*]"	*Developer*	
git fetch	Jenkins	

2.2. Implementasi dan Pengujian Sistem

Berikut akan dipaparkan mengenai implementasi dan pengujian sistem.

2.2.1. Lingkungan Implementasi

Lingkungan implementasi merupakan spesifikasi *hardware*, *software* dan *server* dimana sistem ini akan dipasang dan diakses. Berikut adalah lingkungan implementasinya:

1. Lingkungan Hardware
Spesifikasi perangkat keras yang digunakan adalah sebagai berikut:

Tabel 2.1: Lingkungan Hardware Pembangun perangkat lunak

No	Item	Spesifikasi
1	Prosesor	Intel Celeron N2380 2.4Ghz
2	SSD	128 GB
3	Memory	8 GB
4	Monitor	LCD 11,6”
5	Koneksi Internet	Ada
6	Mouse	Logitech M325
7	Keyboard	*Onboard*

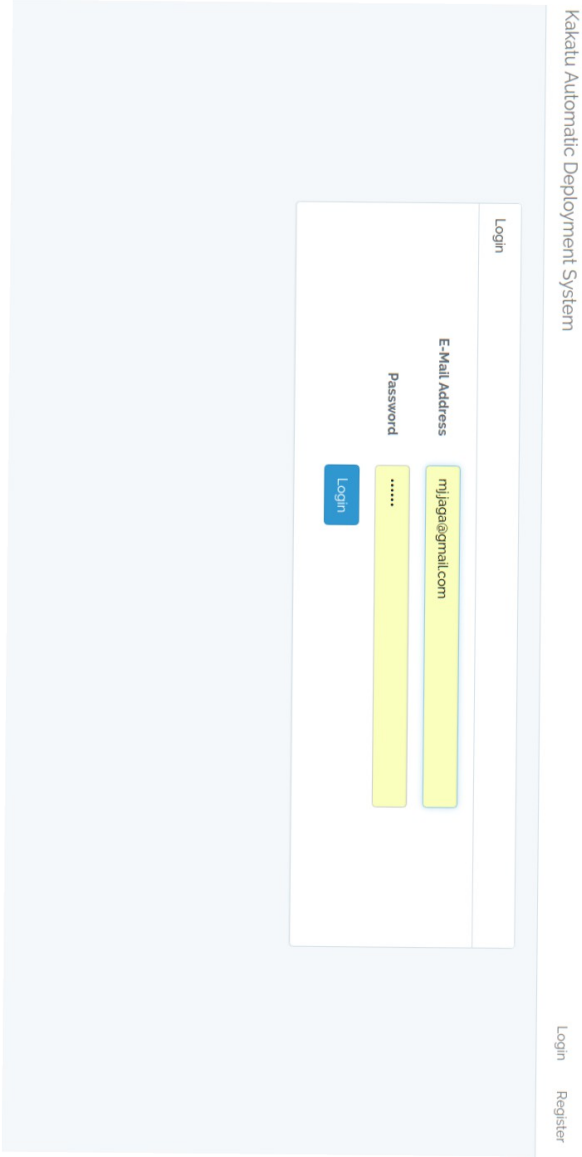
2. Lingkungan Software
Spesifikasi perangkat lunak yang digunakan adalah sebagai berikut:

Tabel 2.2: Lingkungan Software Pembangun perangkat lunak

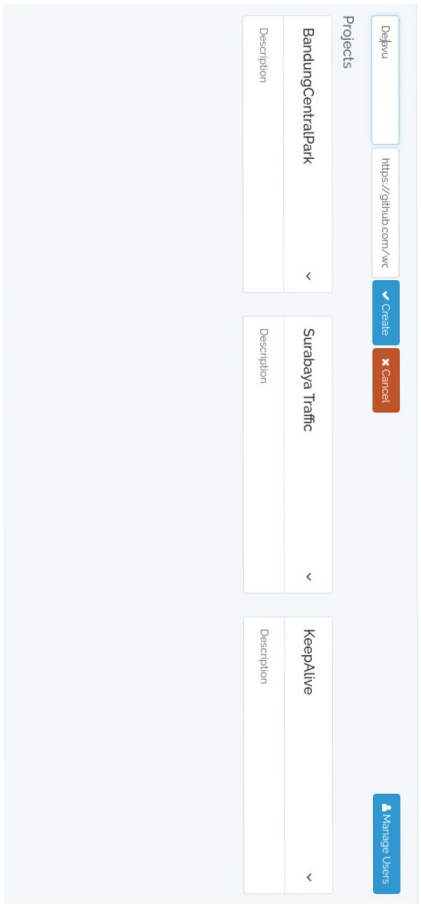
No	Item	Spesifikasi
1	Sistem Operasi	Elementary Loki 0.4
2	Peramban	Google Chrome 56.0.2924.87
3	IDE	Sublime Text Build 3126
4	Bahasa Pemrograman	PHP 7.0 dan Javascript
5	Framework	Laravel 5.0 dan Angular
6	DBMS	Mysql v5.7.17
7	CI Tools	Jenkins v2.32.2
8	Web Server	Apache v2.4.18

2.2.2. Implementasi Antarmuka

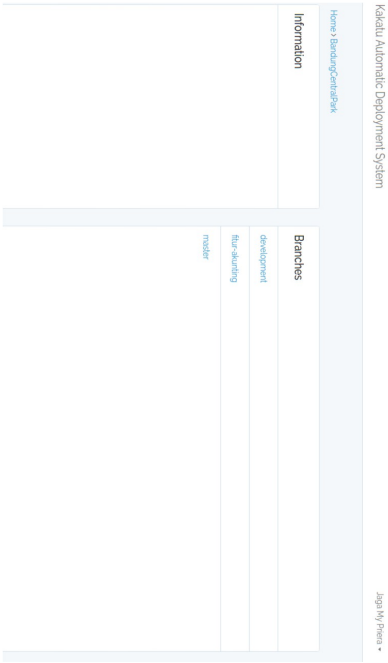
1. Halaman *Login*



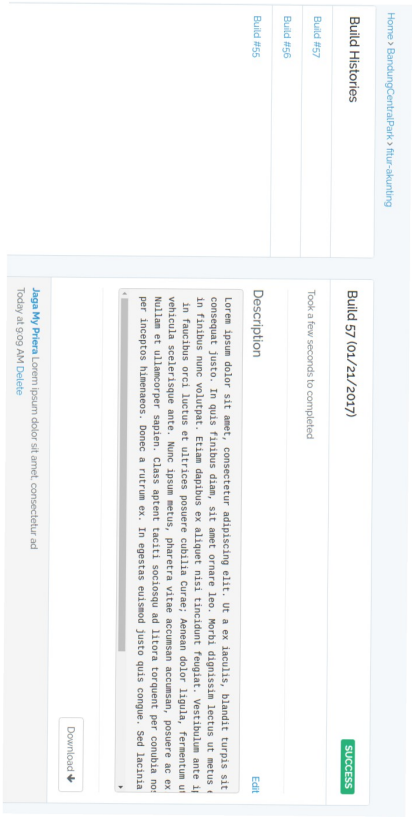
2. Halaman Daftar Proyek



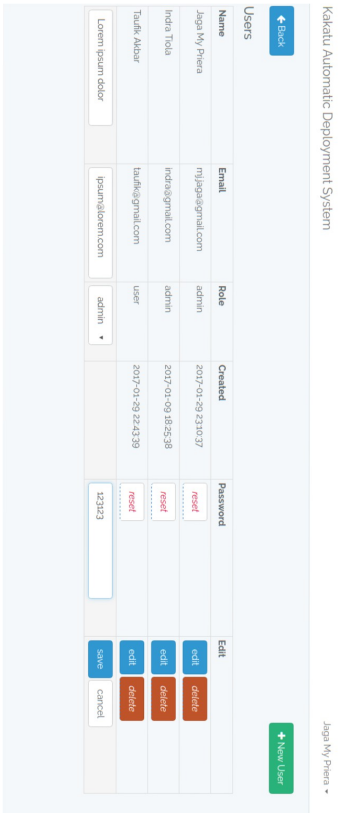
3. Halaman Daftar *Branch*



4. Halaman Detail Riwayat *Build*



5. Halaman Kelola Pengguna



2.2.3. Pengujian

Setelah dilakukan uji coba, didapat perbandingan sistem ini dengan sistem sebelumya:

Kegiatan	Sistem Sebelumya	Sistem Sekarang
Distribusi apk	Mengunggah *File* apk ke Google Drive atau Dropbox lalu membagikan *link* unduh melalui *email*	*File* apk sudah tersedia di CMS ketika kode di-*build*
Komunikasi dan *feedback*	Pemberian *feedback* dan komunikasi dilakukan melalui *email*.	Pemberian *feedback* dan komunikasi dilakukan dalam CMS dan langusng pada versi apk yang terkait.
Dokumentasi	*File* apk dikumpulkan dari Google Drive dan Dropobox. *Feedback* terkait hasil uji coba dikumpulkan melalui *email*	*File* apk dan *feedback* terkait hasil uji coba file telah tersusun menurut versi masing-masing.

3. KESIMPULAN DAN SARAN

Kesimpulan dari pembangunan perangkat lunak ini adalah:

1. Website Automatic Deployment ini dapat mempercepat proses pendistribusian file apk ke tim penguji coba di Kakatu.
2. Website Automatic Deployment ini dapat mempermudah komunikasi antara developer dan tim penguji coba di Kakatu.
3. Website Automatic Deployment ini dapat mempermudah proses dalam mendokumentasikan file apk di Kakatu.

Adapun saran terhadap pengembangan sistem ini adalah:

1. Mengembangkan sistem ini menjadi sebuah layanan yang agar bisa digunakan oleh perusahaan lain.
2. Menambahkan fitur deploy otomatis ke Google Playstore.
3. Mengembangkan aplikasi versi mobile dari sistem ini

4. DAFTAR PUSTAKA

[1] J. Ferguson Smart, *Jenkins - The Definitive Guide*, Creative Commons Edition ed. Wakaleo Consulting, 2011.

[2] S. Chacon and B. Straub, *Pro Git* in *Git - book*, Second Edition ed. Appress, 2014.

[3] R. Miles, Learning UML 2.0, K. Hamilton, Ed. O'Reilly, 2006.

[4] "Pengenalan JSON," dalam *JSON*. [Online]. Available: http://www.json.org/json-id.html. Diakses: 26 Agustus 2016.

[5] Gladwin. B, Ramya. R, dan Rathika. K, "Online Student Resource Management Using HTML5 and Angular -JS," *International Journal of Computer Science and Information Technology Research*, vol. 3, no. 1, hal. 53–57, Jan. 2015.

[6] V. Traynor, "What is the design sprint method and why do I think it's interesting? – Google developers experts," Medium, 2016. [Online]. Available: https://medium.com/google-*developer*-experts/what-is-the-design-sprint-method-and-why-do-i-think-it-s-interesting-3d51d5c37ac1. Accessed: Dec. 5, 2016.

[7] R. Susanto, and A. Dara Andriana, "Perbandingan Model Waterfall dan Prototyping untuk Pengembangan Sistem Informasi," *Majalah Ilmiah UNIKOM*, vol. 14, no. 1, pp. 41–46.

[8] "Google drive - Penyimpanan Awan & Pencadangan *file* untuk Foto, Dokumen & Banyak lagi,". [Online]. Available: https://www.google.com/intl/id/drive/. Accessed: Oct. 26, 2016.

[9] "Buat spreadsheet yang canggih" 2014. [Online]. Available: https://www.google.com/intl/id/sheets/about/. Accessed: Oct. 26, 2016.

[10] Gmail, "Gmail,". [Online]. Available: https://www.google.com/intl/id/mail/help/features.html. Accessed: Oct. 26, 2016.