

E-GOVERNMENT SEMBAKODROID MONITORING HARGA SEMBAKO BERBASIS ANDROID

E-GOVERNMENT SEMBAKODROID MONITORING NINE ESSENTIAL FOOD COMMODITIES PRICES BASED ON ANDROID

¹Ahmad Fahriyannur Rosyady, ²Risnandar, ³Izzatul Ummah

^{1,2}*Program Studi Manajemen Informatika, Fakultas Ilmu Terapan, Universitas Telkom*

³*Program Studi Ilmu Komputasi, Fakultas Teknik, Universitas Telkom*

^{1, 2, 3}*Jl. Telekomunikasi No. 1 Dayeuhkolot, Bandung 40258*

¹fahrickp@gmail.com, ²risnandar@lipi.go.id, ³izzatul.ummah@gmail.com

Naskah diterima : 16 Oktober 2013; Direvisi : 6 Desember 2013 ; Disetujui : 13 Desember 2013

ABSTRAK

Seiring dengan perkembangan zaman dan teknologi yang begitu pesat, banyak manusia yang tidak dapat dipisahkan dari teknologi. Hal ini menyebabkan banyaknya instansi pemerintah yang memanfaatkan teknologi khususnya di bidang Information Technology (IT) dalam menjalankan business process dalam pemerintahan. Dinas Perindustrian dan Perdagangan (Disperindag) Provinsi Jawa Barat merupakan sebuah Dinas yang bertugas untuk mengatur dan mengawasi jalannya arus perdagangan di Jawa Barat. Selain itu, Disperindag Provinsi Jawa Barat juga mempunyai tugas untuk melakukan monitoring (mengawasi) harga sembilan bahan pokok (sembako) yang terdistribusi di pasar-pasar yang berada di Provinsi Jawa Barat. Saat ini, pengelolaan harga sembako hanya dapat dilakukan oleh admin, yang dalam hal ini merupakan pegawai Kementerian Perdagangan sebagai pemantau harga sembako secara nasional. Sedangkan Disperindag Provinsi Jawa Barat memiliki petugas lapangan yang mencatat data langsung dari pasar untuk mendapatkan data harga sembako secara periodik, mulai dari pagi, siang, dan sore hari. Selanjutnya data tersebut dikirimkan ke admin untuk dimasukkan dan dikelola pada server berbasis web untuk disebarluaskan ke masyarakat yang dapat diakses melalui sistem Sembakodroid ini yang dapat diinstal melalui gadget berbasis platform dan Sistem Operasi Android. Sistem Sembakodroid ini terbagi menjadi dua, yaitu mobile system berbasis platform android yang akan digunakan oleh petugas lapangan untuk menginput dan mengolah data laporan harga sembako dan web based system sebagai server dan untuk pembuatan dokumen laporan harga sembako oleh administrator. Metodologi pengembangan sistem ini dengan menggunakan metode Software Development Life Cycle (SDLC) dengan Prototype Model. Dengan adanya sistem monitoring harga sembako Sembakodroid berbasis platform android ini diharapkan dapat membantu petugas lapangan di Disperindag dalam memonitor harga sembako secara periodik dan mempermudah dalam pembuatan laporan data harga sembako secara cepat dan tepat. Selain itu, sistem yang dibangun ini dapat diakses secara mobile oleh petugas lapangan, sehingga proses pengelolaan data harga sembako dapat dilakukan langsung di pasar. Harapan ke depan, sistem ini dapat diadopsi oleh stakeholder lainnya dalam meningkatkan kinerja pelayanan berbasis e-government untuk mencapai good governance.

Kata Kunci: Disperindag, sembako, *android, prototype, e-government, good governance*

ABSTRACT

Currently, government agencies have made use of Information Technology (IT) to implement a business process in government. Department of Industry and Trade (Industry and Trade) of West Java province is responsible for regulating and overseeing the flow of trade in West Java and monitoring (overseeing) the price of nine essentials food commodities (sembako) that is distributed in the markets around West Java Province. Currently, management of nine essentials food commodities prices can only be done by an admin, which in this case are employees of the Ministry of Commerce as food prices nationwide monitor. While Disperindag West Java province have field workers who recorded the data directly from the market to get food commodities prices data periodically, ranging from morning, afternoon, and evening. Furthermore, the data is sent to the admin and managed to put on a web-based server for dissemination to the public which can be accessed through this Sembakodroid system that can be installed through platform-based gadgets and Android Operating System. Sembakodroid system is divided into two, the Android mobile platform-based system that will be used by field personnel to input and process data reports food prices and a web-based system as the server untuk melakukan laporan dokumen dari harga pangan oleh administrator. With the nine essentials food commodities prices Sembakodroid monitoring system based on android platform is expected to assist officers in the field of Industrial and Trade Agencies to monitor food prices periodically and facilitate in making food prices report the data accurately and quickly intervening. In addition, the system built in a mobile accessible by field workers, so that the food prices data management process can be performed directly in the market. For the next, this system can be adopted by other stakeholders to improving performance of e-government services based in order to achieve good governance.

Keywords: *Disperindag, sembako, android, prototype, e-government, good governance*

LATAR BELAKANG

Tujuan

Dinas Perindustrian dan Perdagangan (Disperindag) provinsi Jawa Barat (Jabar) merupakan sebuah Dinas yang bertugas untuk mengatur dan mengawasi jalannya arus perdagangan di Jawa Barat. Selain itu, Disperindag Jabar mempunyai tugas untuk melakukan monitoring (mengawasi) harga sembilan bahan pokok (Sembako) yang ada di provinsi Jawa Barat. Monitoring harga sembako ini mencakup beberapa hal di antaranya : Penginputan data, dan pemantauan sembako pada pasar-pasar

diseluruh Jawa Barat. Saat ini pengelolaan harga sembako hanya dapat dilakukan oleh admin dalam hal ini merupakan pegawai kementerian perdagangan. Disperindag Jabar memiliki petugas lapangan yang mencatat data langsung dari pasar untuk mendapatkan data harga sembako. Kemudian data tersebut dikirimkan ke admin untuk dimasukkan dan dikelola pada website. Hal ini berdampak pada lamanya waktu penyampain informasi harga sembako kepada masyarakat. Selain itu, terdapat beberapa pasar dengan nama yang sama, namun berbeda lokasi.

Berdasarkan permasalahan tersebut, maka perlu dilakukan penelitian dan dibangun sebuah sistem yang dapat digunakan oleh petugas lapangan dan admin untuk mengelola dan membuat laporan data harga sembako. Selain itu, sistem yang dibangun dapat diakses secara *mobile* oleh petugas lapangan, sehingga proses pengelolaan data harga sembako dapat dilakukan langsung di pasar. Dengan dibangunnya sistem tersebut diharapkan dapat menjadi sarana penunjang kinerja Disperindag Jabar dalam mengelola data harga sembako dan menyampaikannya kepada masyarakat. Berdasarkan latar belakang yang telah ditulis, ada beberapa rumusan masalah yaitu : Bagaimana menentukan pusat perdagangan komoditas sembako tertentu dengan sistem yang terkomputerisasi dengan berbasis *mobile*; Bagaimana menentukan suatu daerah yang mempunyai potensi penjualan paling banyak terhadap suatu komoditas tertentu; dan Bagaimana cara pembuatan laporan harga sembako secara periodik. Sedangkan tujuan dari penelitian ini diantaranya : Membangun sistem yang dapat menentukan pusat perdagangan komoditas sembako tertentu; Membangun sistem yang dapat menentukan suatu daerah yang mempunyai potensi penjualan paling banyak terhadap suatu komoditas tertentu; dan Membangun sistem pembuatan harga sembako secara periodik.

Berdasarkan beberapa uraian sebelumnya, maka didapatkan batasan masalah, di antaranya : sistem ini hanya me-monitoring harga sembako; sistem ini berbasis Android; Data harga sembako ditampilkan secara statistik; Laporan dibuat dalam bentuk file spread sheet; dan Batasan kabupaten/kota yang berada di provinsi Jawa Barat berdasarkan komoditas sembako utama yang ada pada daerah tersebut.

TINJAUAN PUSTAKA

Aplikasi

Aplikasi adalah seperangkat intruksi khusus dalam komputer yang dirancang agar kita dapat menyelesaikan tugas-tugas tertentu. Sebagai contoh, aplikasi word processing adalah sebuah aplikasi yang diperuntukkan membuat dokumen tertulis, aplikasi web browser adalah aplikasi yang diperuntukkan untuk mencari sesuatu dan menampilkan halaman web (Shelly, 2009).

Sembako

Sembako adalah singkatan dari sembilan bahan pokok yang terdiri atas berbagai bahan-bahan makanan dan minuman yang secara umum sangat dibutuhkan masyarakat Indonesia. Tanpa sembako kehidupan rakyat Indonesia bisa terganggu karena sembako merupakan kebutuhan pokok utama sehari-hari yang wajib ada dijual bebas di pasar (Hambali,

1998). Di bawah ini adalah daftar nama anggota bahan pokok sembako sesuai dengan keputusan Menteri Industri dan Perdagangan NO. 115/MPP/KEP/2/1998 tanggal 27 Februari 1998, yaitu antara lain : Beras dan sagu; Jagung; Sayur-sayuran dan buah-buahan; Daging (Sapi dan Ayam); Susu; Gula Pasir; Garam yang mengandung yodium/iodium; Minyak garam dan Margarin; dan Minyak Tanah atau Gas Elpiji.

Android

Android adalah sebuah mobile operating system yang bersifat terbuka dan berbasis pada Linux operating system. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka android dipuji sebagai "*The first mobile open source platform*" android merupakan generasi baru platform mobile, platform yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. sistem oprasi android dilisensikan dibawah GNU, *General public Licensi* Versi 2 (GPLv2) (Syafaat, 2011). Aplikasi android dapat dikembangkan pada sistem operasi berikut : Windows XP Vista/Seven; Mac OS X (Mac OS X 10.4.8 atau lebih baru; LINUX;

Object Oriented Programming (OOP)

Object Oriented Programming (OOP) atau pemrograman berorientasi objek adalah konsep pemrograman yang difokuskan pada penciptaan kelas yang merupakan abstraksi/prototype dari suatu objek. Kelas ini harus mengandung sifat (data) dan tingkah laku (*method*) umum yang dimiliki oleh objek-objek yang kelak akan dibuat (diinstansiasi). Data dan *method* merupakan anggota (*member*) dari suatu kelas. Objek adalah kesatuan entitas (benda), baik yang berwujud nyata ataupun hanya suatu sistem atau konsep yang memiliki sifat karakteristik dan fungsi. Kelas adalah pemodelan dari objek yang berisi informasi (aturan) tentang sifat karakteristik (data) dan tingkah laku (*method*) yang dimiliki oleh objek tersebut (Hermawan, 2004). Dalam pemrograman berorientasi objek terdapat beberapa fitur antara sebagai berikut :

1. Enkapsulasi (*Encapsulation*)

Enkapsulasi adalah suatu cara untuk menyembunyikan detail implementasi detail dari sebuah kelas. Terdapat dua hal mendasar dari enkapsulasi yaitu : *Information Hiding* (penyembunyian detail dari atribut dan method pada sebuah kelas) dan Interface untuk pengaksesan data : suatu method untuk mengambil, memberikan atau mengubah suatu nilai.

2. Abstraksi (*Abstraction*)

Arti abstraksi mengacu kepada atribut dari sebuah objek yang membedakan antara satu objek dengan objek yang lain. Dalam pemrograman berorientasi objek konsep ini berada pada pembuatan kelas. Sebenarnya kelas tidak memiliki data, tetapi sebuah objeklah yang akan menyimpan data.

3. Pewarisan (*Inheritance*)

OOP mengizinkan untuk menciptakan *class* dari *class* lain yang telah ada inilah yang disebut *inheritance*. Karena pada proses penciptaan sebuah *class* dengan cara menurunkan objek dari *class* lainnya, maka objek tersebut otomatis mewarisi semua properti dan *method* dasar dari *class*. Dari sinilah nanti akan muncul istilah *parent class* dan *child class*. *Parent class* atau *superclass* adalah *class* yang menjadi master dari turunannya (Komputer, 2010). Dalam pemrograman berorientasi objek, terdapat beberapa jenis pewarisan, yaitu : *Single Inheritance* (hanya terdapat satu *superclass*) dan *Multiple Inheritance* (terdapat lebih dari satu *superclass*). Dalam Java sendiri konsep tentang *Multiple Inheritance* dihapuskan, karena kompleksitas dari konsep tersebut.

4. Polimorfisme (*Polymorphism*)

Polimorfisme adalah sebuah fitur yang memungkinkan kita untuk memberikan arti atau penggunaan yang berbeda bagi

sebuah entitas dalam konteks yang berbeda-beda. Entitas tersebut bisa berupa variabel, *method*, atau sebuah objek. Polimorfisme bisa digunakan sebagai kategori umum bagi sebuah entitas dalam tindakan yang berbeda-beda (Putro, R, & Rasiana, 2009).

HTML

Hypertext Markup Language (HTML) adalah sebuah *script* pemrograman yang mengatur bagaimana kita menyajikan informasi di dunia internet dan bagaimana informasi itu membawa kita melompat dari satu tempat ke tempat lainnya (Interprise, 2011).

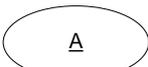
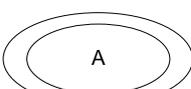
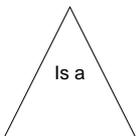
Beberapa kelebihan yang dijanjikan pada HTML: Dapat ditulis dalam sintaks HTML (dengan tipe media text/HTML) dan XML; Integrasi yang lebih baik dengan aplikasi situs dan pemrosesannya; Integrasi ('inline') dengan *doctype* yang lebih sederhana; Penulisan kode yang lebih efisien; dan Konten yang ada di situs lebih mudah terindeks oleh *search engine*.

Entity Relationship Diagram (ERD)

ER adalah salah satu pemodelan basis data konseptual yang menggambarkan basis data ke dalam bentuk entitas-entitas dan relasi yang terjadi di antara entitas-entitas yang ada. Entitas diartikan sebagai 'objek' di dunia nyata yang bisa dibedakan dengan 'objek' yang lain. Relasi diartikan sebagai

hubungan yang terjadi di antara satu entitas dengan entitas yang lainnya (Rohim, 2009).

Tabel 1. *Entity Relationship Diagram (ERD)*

Simbol	Deskripsi
	Entitas
	Atribut
	Atribut kunci primer
	Atribut multivalai
	Relasi
	Spesialisasi/ generalisasi

Tujuan dibuatnya ERD adalah untuk memudahkan penerjemahan hubungan antar tabel, supaya dalam penyusunannya tidak terjadi kesimpangsiuran tabel yang saling berhubungan. Beberapa aturan untuk mentransformasi ERD ke dalam tabel yaitu : Setiap himpunan entitas menjadi tabel baik himpunan entitas kuat atau lemah; Setiap atribut menjadi kolom di tabel; dan Kardinalitas relasi akan menentukan jumlah tabel yang terbentuk. Kardinalitas relasi dari himpunan entitas yang saling memiliki relasi akan menentukan banyaknya tabel yang bisa dibuat. Adapun aturannya sebagai berikut :

1. Relasi dengan derajat 1-1
Pilih primary key di 1 himpunan entitas untuk menjadi foreign key bagi himpunan entitas yang lain.
2. Relasi dengan derajat 1-N atau N-1
Primary key pada tabel berkardinalitas sedikit menjadi foreign key pada tabel berkardinalitas banyak.
3. Relasi dengan derajat N-N
Primary key pada 2 himpunan entitas yang berhubungan menjadi foreign key pada tabel baru.

MY SQL

SQL muncul pertama kali diawal tahun 1970-an sebagai sebuah hasil penelitian pada lab IBM, San Joe, California yang dilakukan oleh Donald C Meserly dan Raymond F Boyce. Versi pertama ini diberikan nama SEQUEL (Structure English Query Language). Rencana awalnya SEQUEL ini akan digunakan dalam DB2 sebagai sebuah produk RDBMS dari IBM. Kemudian IBM memberikan nama lain pada SEQUEL ini dengan nama SQL (Rohim, 2009). MySQL adalah sebuah perangkat lunak sistem manajemen data SQL atau dikenal dengan DBMS (Database Management System), database ini multi-thread dan multi-user. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structure Query

Language). Sebagai databaseserver, MySQL dapat dikatakan lebih unggul dibanding database server lainnya, terutama dalam kecepatan (M Huda, 2011).

Berikut beberapa keistimewaan MySQL, antara lain: Portability, MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, dan masih banyak lagi; Multi-user, MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik; Skalabilitas dan *limits*, MySQL mampu menangani database dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya; Security, MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

JavaScript

JavaScript adalah bahasa scripting yang didesain untuk menambah interaktif suatu web. Javascript cukup populer di internet dan dapat bekerja di sebagian besar browser populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode javascript dapat disisipkan dalam halaman web menggunakan tagscript. Javascript

merupakan bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi) dan baris kodenya dijalankan di komputer (web browser) karena biasanya disisipkan (*embedded*) dalam halaman HTML (Sunyoto, 2007).

CSS (Cascading Style Sheets)

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. CSS diperkenalkan untuk pengembangan website pada tahun 1996. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C). CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis-suara atau pembaca layar), dan juga alat pembaca braille. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS (Suryatiningsih, 2009). Keuntungan menggunakan CSS : Memisahkan *presentation* sebuah dokumen dari *content document* itu sendiri; Mempermudah dan mempersingkat

pembuatan dan pemeliharaan dokumen web; dan Mempercepat proses *rendering* /pembacaan HTML.

PHP

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada server. sedangkan yang dikirimkan ke browser hanya hasilnya saja. Ketika seorang pengguna internet membuka suatu situs yang menggunakan fasilitas *server side scripting* PHP, maka terlebih dahulu server yang bersangkutan akan memproses semua perintah PHP di server lalu mengirimkan hasilnya dalam format HTML ke web server pengguna internet tadi. Sehingga kode asli yang ditulis dengan PHP tidak terlihat di browser pengguna. PHP merupakan *software* yang open source. PHP juga dapat berjalan lintas *platform*, yaitu dapat digunakan dengan sistem operasi (Windows dan Linux) dan web server apapun, misalnya: PWS, IIS, dan Apache (Sutanta, 2004).

Class Diagram

Class diagram ialah diagram yang memperlihatkan himpunan kelas kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relevansi-relevansi. Diagram

kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

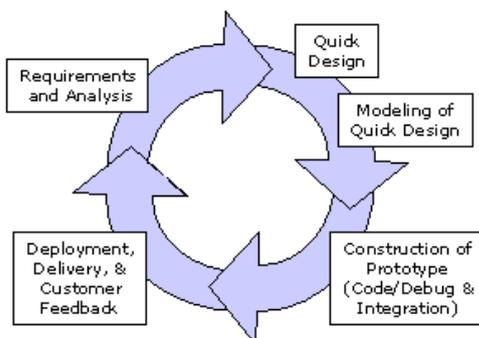
Flowmap

Flowmap merupakan diagram yang menggambarkan aliran dokumen pada suatu prosedur kerja di organisasi. Penggambaran biasanya diawali dengan mengamati dokumen apa yang menjadi media data atau informasi. Selanjutnya ditelusuri bagaimana dokumen tersebut terbentuk, dan mengalir, perubahan apa yang terjadi pada dokumen tersebut, proses apa yang terjadi terhadap dokumen tersebut, dan seterusnya. (Soeherman, 2008).

METODOLOGI

Metodologi penelitian yang digunakan adalah metode *prototyping*. Metode *prototyping* merupakan sebuah metode yang digunakan untuk pembangunan sebuah proyek IT dimana klien hanya memberikan beberapa kebutuhan umum software tanpa detail input, proses, atau detail output. Di lain waktu mungkin tim *developer* tidak yakin terhadap efisiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan form *user interface*. Ketika situasi seperti ini, *prototype paradigm* menawarkan pendekatan terbaik (Pressman, 2010).

Model *prototype* memiliki beberapa tahapan yaitu pengumpulan kebutuhan, membangun *prototype*, evaluasi *prototype*, mengkodekan sistem, menguji sistem, evaluasi sistem, dan menggunakan sistem. Penjelasan dari tahap-tahapan tersebut adalah sebagai berikut :



Gambar 1. *Prototyping Model*

Requirements and Analysis

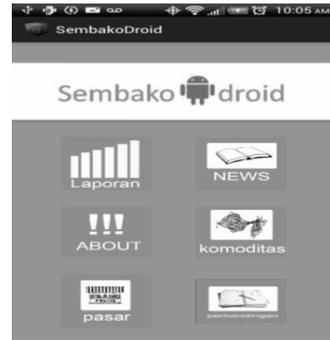
Pada tahap ini dilakukan pengumpulan data-data yang dibutuhkan dengan cara melakukan wawancara kepada Disperindag Jabar, dan melakukan observasi di lapangan untuk mengamati secara langsung proses pengolahan data di Disperindag Jabar.

Quick Design

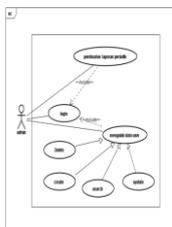
Berdasarkan data yang telah dikumpulkan, selanjutnya akan dibuat perancangan sistem antara lain, flowmap sistem yang sedang berjalan dan sistem yang akan dibangun, *use case diagram*, *Entity Relationship Diagram (ERD)*. Berikut tahapan *quick design* yang dilakukan dalam penelitian ini.



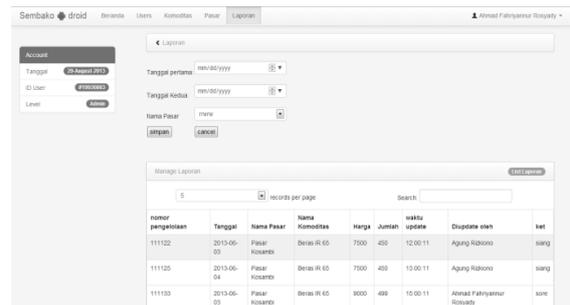
Gambar 2. Use Case Diagram Mobile System



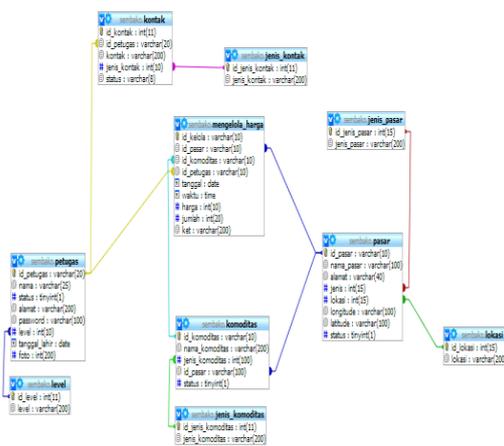
Gambar 5. Interface Design Android Platform



Gambar 3. Use Case Diagram Web-based System



Gambar 6. Interface Design Web-based System



Gambar 4. ERD Database System

Construction of Prototype

Setelah perancangan sistem telah disetujui, selanjutnya diterjemahkan perancangan sistem dan tampilan sistem ke dalam bahasa pemrograman. Berikut surat keterangan persetujuan untuk melakukan inovasi dan penilitan dari beberapa *stakeholder* yang terlibat.

Modelling of Quick Design

Pada tahap ini diperlihatkan hasil perancangan sistem kepada Disperindag Jabar. Berikut *interface design* dalam penelitian Sembakodroid ini.



Gambar 7. Surat Keterangan Disperindag



Gambar 8. Surat Keterangan Bandung Techno Park Selaku Stakeholder dalam *Business Innovation Center*

Deployment, Delivery and customer feedback

Setelah program selesai dibuat, maka program tersebut harus di-test untuk mengetahui apakah program tersebut sudah benar-benar sesuai dengan kebutuhan atau belum. Pengujian program menggunakan metode *black box testing*. Pengujian sistem

sembakodroid dilakukan dengan model *black box testing* yang merupakan sebuah pengujian yang dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsionalitas dari *interface* perangkat lunak. Pada pengujian *black box testing*, pengujian dilakukan oleh *user* dan dikomunikasikan kembali beberapa kebutuhan yang diperlukan *user* sejak awal *user* dan *developer* melakukan identifikasi sistem. Untuk tahapan pengujian, terlampir dalam Bagian Pengujian dalam tulisan ini.

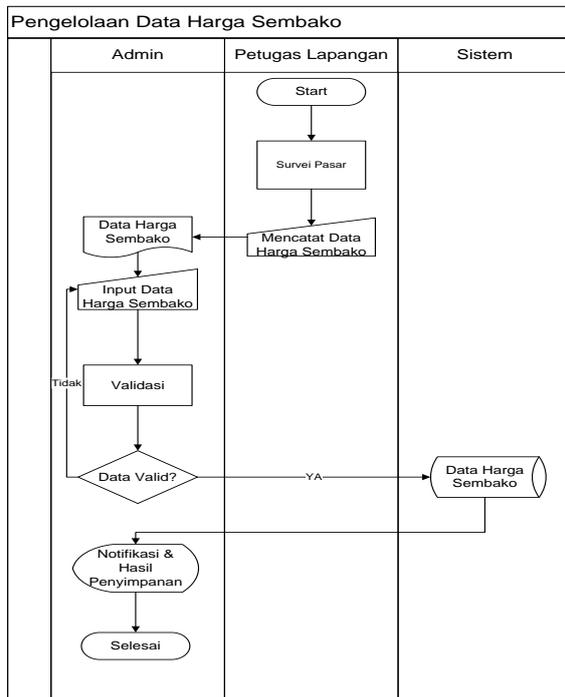
ANALISIS DAN PERANCANGAN

Kondisi Saat Ini

Disperindag Jabar merupakan sebuah dinas yang bertugas melakukan *monitoring* harga sembako. Kondisi saat ini adalah :

1. Petugas lapangan melakukan survei pasar dan mencatat data harga sembako kemudian data tersebut diserahkan kepada *admin* dalam bentuk dokumen.
2. Langkah selanjutnya *admin* menginputkan data harga sembako sesuai data yang telah diberikan oleh petugas lapangan.
3. Langkah selanjutnya sistem memvalidasi kebenaran penginputan data yang dilakukan oleh *admin*.
4. Jika benar, data disimpan ke *database* data harga sembako dan muncul notifikasi data yang diinputkan benar, jika data yang dimasukkan salah maka

muncul notifikasi bahwa data yang diinputkan salah.



Gambar 9. Flowmap Kondisi Saat Ini

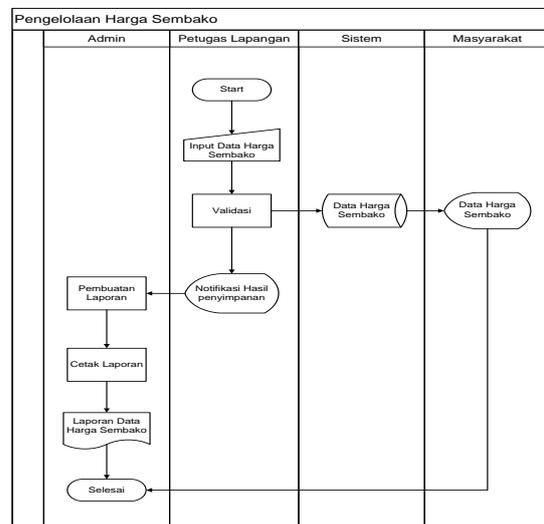
Sistem yang Diusulkan

Aplikasi sembakodroid merupakan sebuah aplikasi yang memonitoring harga sembako berbasis android yang digunakan untuk memudahkan penggunanya mengakses harga sembako melalui gadget android. Beberapa manfaat dari aplikasi ini adalah :

1. Petugas lapangan melakukan survei pasar dan menginputkan data harga sembako kemudian data tersebut divalidasi oleh sistem, jika data yang diinputkan benar maka data disimpan di database data harga sembako dan muncul notifikasi bahwa data berhasil disimpan. Jika data yang diinputkan salah maka data tidak di simpan di

database dan muncul notifikasi bahwa data yang diinputkan salah.

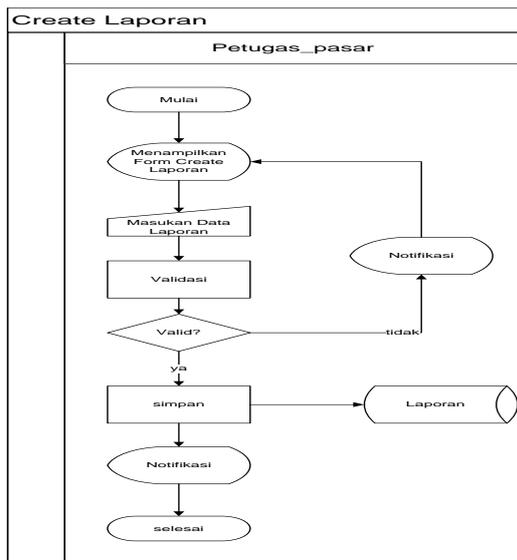
2. Masyarakat dapat melihat data harga sembako yang telah tersimpan dalam database data harga sembako.
3. Admin dapat melakukan pembuatan laporan secara periodik sesuai data yang telah tersimpan dalam database data harga sembako. Kemudian laporan tersebut dapat dicetak dalam sebuah dokumen.



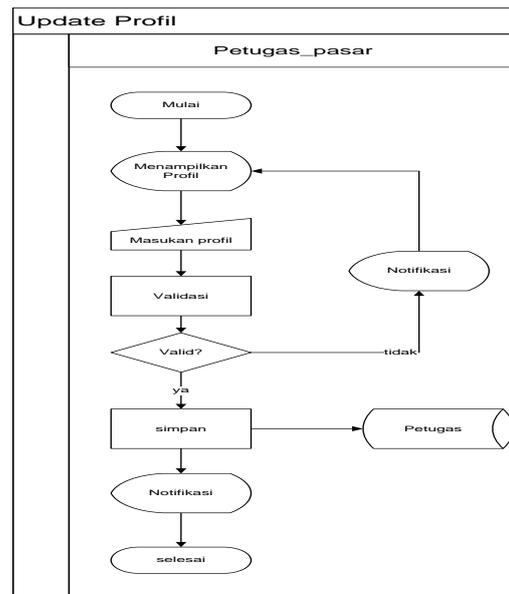
Gambar 10. Flowmap Sistem yang Diajukan

1. Flowmap create laporan

Flowmap create laporan digunakan untuk menggambarkan proses create laporan harga sembako yang dilakukan oleh petugas Lapangan



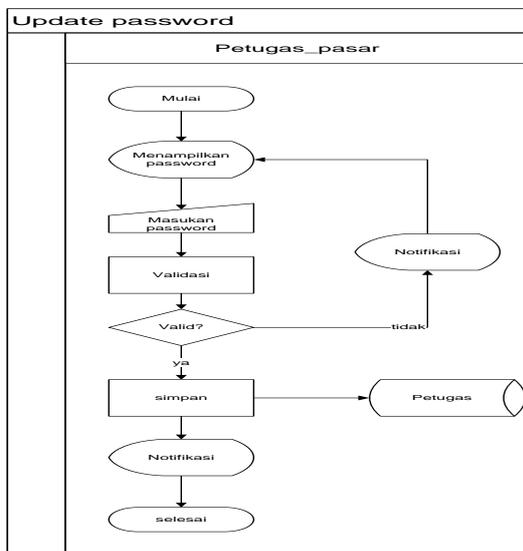
Gambar 11. Flowmap Create Laporan



Gambar 13. Update Profil

2. Flowmap updatepassword

Flowmap update password adalah flowmap yang digunakan untuk mendefinisikan update password petugas lapangan pada android.



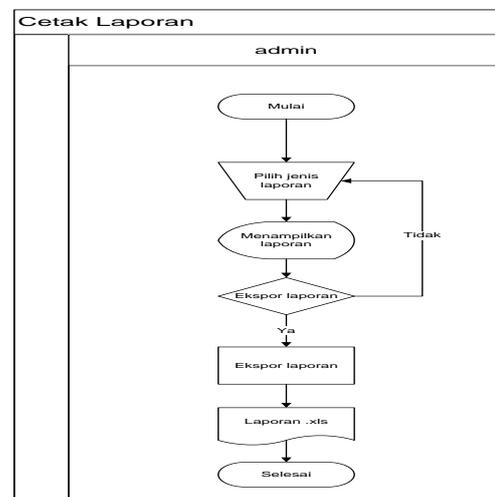
Gambar 12. Flowmap Update Password

3. Flowmap update profil

Flowmap update profil adalah flowmap yang digunakan untuk menggambarkan update profil petugas lapangan yang ada di android.

4. Flowmap cetak laporan

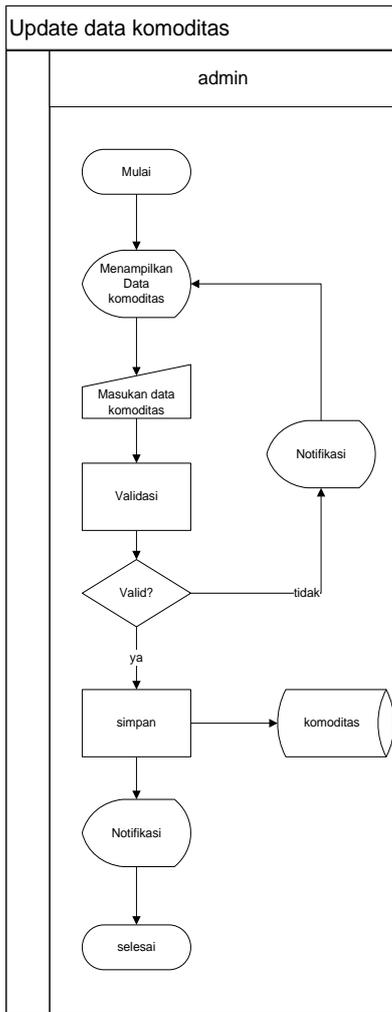
Flowmap cetak laporan adalah flowmap yang digunakan untuk menggambarkan proses cetak laporan yang ada pada website.



Gambar 14. Flowmap Cetak Laporan

5. Flowmap update data komoditas

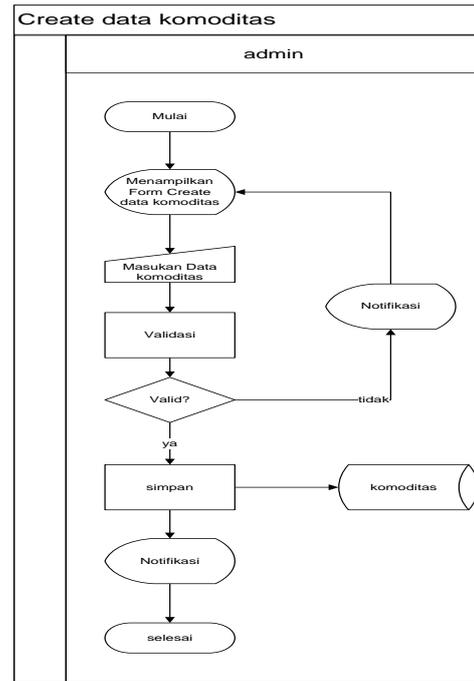
Flowmap update data komoditas adalah flowmap yang digunakan untuk menggambarkan proses update data komoditas yang ada pada website.



Gambar 15. Flowmap Update Data Komoditas

6. Flowmap create data komoditas

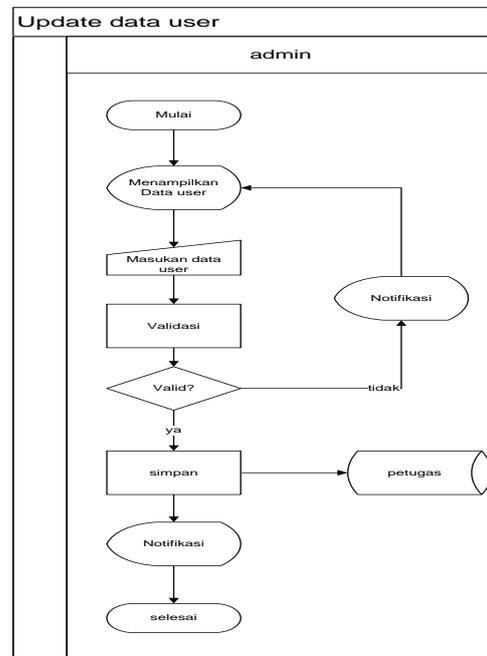
Flowmap create data komoditas adalah flowmap yang digunakan untuk menggambarkan proses create data komoditas yang ada pada website.



Gambar 16. Flowmap Create Data Komoditas

7. Flowmap update data user

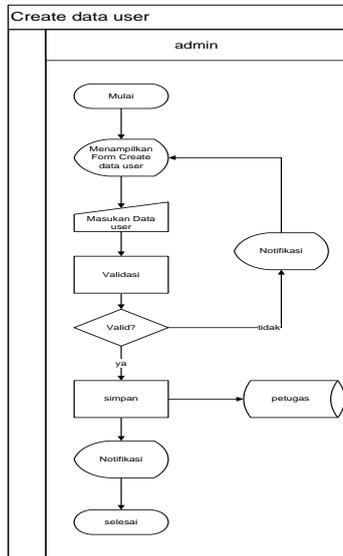
Flowmap Update data user adalah flowmap yang digunakan untuk update data user yang ada pada website.



Gambar 17. Update Data User

8. Flowmap create data user

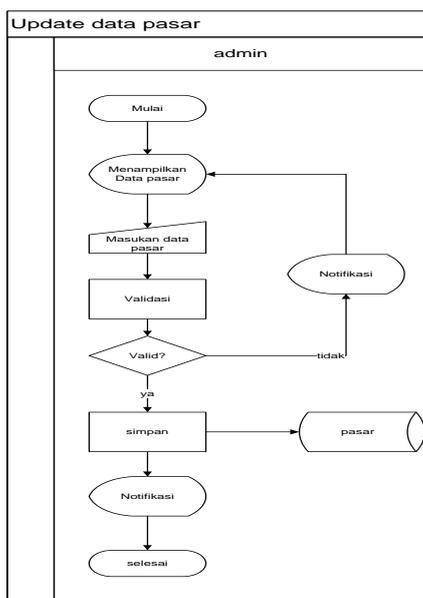
Flowmap create data user adalah flowmap yang digunakan untuk menggambarkan create data user pada website.



Gambar 18. Create Data User

9. Update data pasar

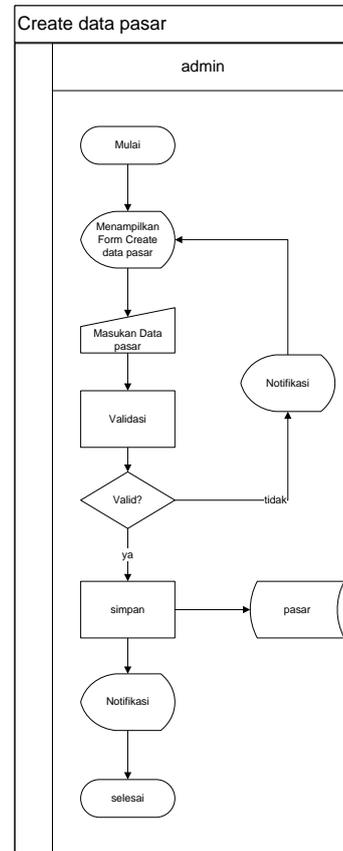
Flowmap update data user adalah flowmap yang digunakan untuk menggambarkan update data user pada website.



Gambar 19. Update Data User

10. Create Data User

Flowmap create data user adalah flowmap yang digunakan untuk menggambarkan create data user pada website.



Gambar 20. Create Data Pasar

Desain Produk

Aplikasi sembakodroid memiliki 3 aktor yaitu *admin*, *petugas lapangan* dan *masyarakat*. yang mempunyai hak akses masing-masing oleh karena itu pada sub bab desain produk ini di jelaskan tentang gambaran aktor serta proses bisnis didalamnya. aplikasi ini mempunyai 2 use case yaitu use case untuk aplikasi *mobile* dan use case untuk aplikasi *website*.

Tabel 8. *Use Case Login*

Aktor	Sistem
1. Memasukkan <i>username</i> dan <i>password</i> .	
2. Menekan tombol <i>login</i> .	
	3. Validasi <i>username</i> dan <i>password</i> .
	4. Jika salah, kembali ke no. 1 dan menampilkan pesan kesalahan.
	5. Jika benar, tampil halaman utama aplikasi.

b. Use Case Update data User

Use case : *Update data user*

Actor : Petugas lapangan

Description : Digunakan untuk meng *update data user*

Pre-condition : Muncul *formupdate data user*

Post-condition: *Data user* berhasil disimpan

Tabel 9. *Update Data User*

Aktor	Sistem
1. Memasukkan <i>data user</i>	
2. Menekan tombol "OK"	
	3. Muncul <i>pop up</i>
4. Jika memilih batal kembali ke nomor 1	
5. Jika memilih "OK" lanjut ke proses validasi	
	6. Validasi data
	7. Jika data salah kembali ke nomor langkah nomor 1
	8. Jika data benar, maka data baru

Aktor	Sistem
	akan tersimpan di database data <i>user</i>

c. Use Case Create Data User

Use case : *Create data user*

Actor : Petugas lapangan

Description : Digunakan untuk meng-*create data user*

Pre-condition : Muncul *formcreate data user*

Post-condition : *Data user* berhasil disimpan

Tabel 10. *Create Data User*

Aktor	Sistem
1. Memasukkan <i>data user</i>	
2. Menekan tombol "OK"	
	3. Muncul <i>pop up</i>
4. Jika memilih batal kembali ke nomor 1	
5. Jika memilih "OK" lanjut ke proses validasi	
	6. Validasi data
	7. Jika data salah kembali ke nomor langkah nomor 1
	8. Jika data benar, maka data baru akan tersimpan di database <i>user</i>

d. Use Case delete data user

Use case : *Delete data user*

Actor : Petugas lapangan

Description : Digunakan untuk men-*delete data user*

Pre-condition : Muncul *data user*

Post-condition: *Data user* berhasil di-*delete*

Tabel 11. Use Case Delete Data User

Aktor	Sistem
	1. Sistem menampilkan data user
2. Memilih data yang akan di-delete	
3. Menekan tombol delete	
4. Jika memilih batal kembali ke nomor 1	
	5. Muncul <i>pop up</i> , konfirmasi data yang akan di-delete
6. Jika memilih "OK" maka data berhasil di-delete, jika memilih "Batal" maka kembali ke nomor 1	
	7. Muncul <i>pop up</i> data berhasil di-delete
	8. Data yang telah di-delete disimpan di database <i>history</i>

e. Use Case Search data user

Use case : Search data user
Actor : Petugas lapangan
Description : Digunakan untuk search data user
Pre-condition : Muncul *field search data user*
Post-condition : search data user berhasil ditampilkan

Tabel 12. Use Case Search Data User

Aktor	Sistem
	1. sistem menampilkan <i>field search data user</i>

Aktor	Sistem
2. Memasukkan <i>field</i> pencarian	
3. Menekan tombol search	
	4. Muncul data hasil pencarian

f. Use Case Update data laporan

Use case : Update data laporan
Actor : Petugas lapangan
Description : Digunakan untuk meng-update data laporan
Pre-condition : Muncul form *update data laporan*
Post-condition: Data laporan berhasil disimpan

Tabel 13. Use Case data laporan

Aktor	Sistem
1. Memasukkan inputan data laporan	
2. Menekan tombol "OK"	
	3. Muncul <i>pop up</i>
4. Jika memilih batal kembali ke nomor 1	
5. Jika memilih "OK" lanjut ke proses validasi	
	6. Validasi data inputan
	7. Jika data salah kembali ke nomor langkah no 1
	8. Jika data benar, maka data baru akan tersimpan di database data laporan

g. Use Case create data laporan

Use case : Create data laporan
Actor : Petugas lapangan

Description : Digunakan untuk meng-create data laporan
Pre-condition : Muncul form create data laporan
Post-condition : Data laporan berhasil disimpan

Tabel 14. Use Case Create Data Laporan

Aktor	Sistem
1. memasukkan inputan data laporan 2. menekan tombol "OK"	3. muncul <i>pop up</i>
4. jika memilih batal kembali ke nomor 1 5. jika memilih "OK" lanjut ke proses validasi	6. validasi data inputan 7. jika data salah kembali ke nomor langkah no 1 8. Jika data benar, maka data baru akan tersimpan di database laporan

h. Use Case Delete Data Laporan

Use case : Delete data laporan
Actor : Petugas lapangan
Description : Digunakan untuk men- delete data laporan
Pre-condition : Muncul data laporan
Post-condition : Data laporan berhasil di-*delete*

Tabel 15. Delete Data Laporan

Aktor	Sistem
2. Memilih data yang akan di- <i>delete</i> 3. Menekan tombol delete	1. sistem menampilkan data laporan

Aktor	Sistem
4. jika memilih batal kembali ke nomor 1	4. Muncul <i>Pop up</i> , konfirmasi data yang akan di- <i>delete</i>
5. jika memilih "OK" maka data berhasil di- <i>delete</i> , jika memilih "Batal" maka kembali ke nomor 1.	6. Muncul <i>Pop up</i> data berhasil di- <i>delete</i> 7. Data yang telah di- <i>delete</i> disimpan di database history

i. Use Case Search Data Laporan

Use case : Search data laporan
Actor : Petugas lapangan
Description : Digunakan untuk men-*search* data laporan
Pre-condition : Muncul *field search* data

Tabel 16. Use Case Search Data

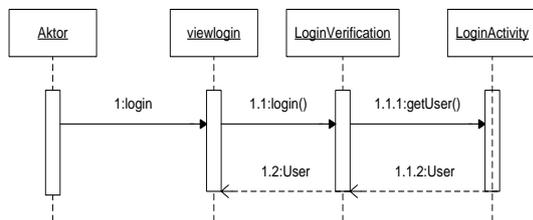
Aktor	Sistem
2. Memasukkan <i>field</i> pencarian 3. Menekan tombol search	1. Sistem menampilkan <i>field search</i> data laporan 4. Muncul data hasil pencarian

2. Sequence Diagram

1. Sequence Diagram Login

Sequence digram login memaparkan tentang proses bisnis login aplikasi pada

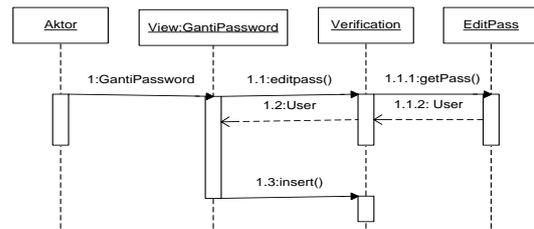
aplikasi *mobile*. Aktor melakukan login ke aplikasi kemudian muncul *interface* untuk *login* ke aplikasi. Setelah aktor *input username* dan *password* nya maka sistem akan memvalidasi data *inputan* user. Jika benar maka class *loginActivity* akan memberikan izin kepada aktor untuk masuk ke aplikasi, jika salah maka aktor diharuskan untuk memasukkan *username* dan *password* yang benar.



Gambar 22. Sequence Diagram Login

2. Sequence Diagram Edit Password

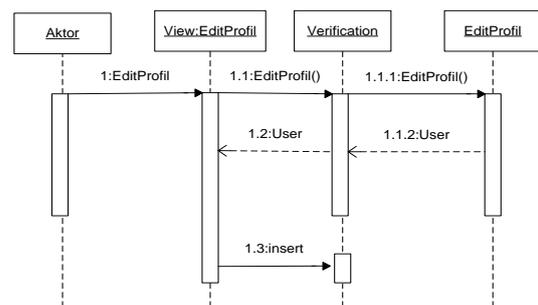
Sequence digram *edit password* menggambarkan tentang proses bisnis ganti *password* pada aplikasi *mobile*. pada proses ini aktor memilih fungsionlitas *edit password* kemudian aplikasi memunculkan *interfaceeditpassword*. setelah *interface* muncul, maka aktor diminta untuk memasukkan *password* lama dan *password* barunya. kemudian sistem akan meverifikasi inputan tersebut, selanjutnya class *edit pass* akan memberikan hasil eksekusi berupa *password* berhasil di ganti atau tidak.



Gambar 23. Sequence Diagram Ganti Password

3. Sequence Diagram Edit Profil

Sequence digram *edit profil* menggambarkan tentang proses bisnis *edit profil* pada aplikasi *mobile*. pada proses ini aktor memilih fungsionlitas *edit profil* kemudian aplikasi memunculkan *interface edit password*. setelah *interface* muncul, maka aktor diminta untuk memasukkan profil barunya. kemudian sistem akan meverifikasi inputan tersebut, selanjutnya class *edit profil* akan memberikan hasil eksekusi berupa profil berhasil di ganti atau tidak.

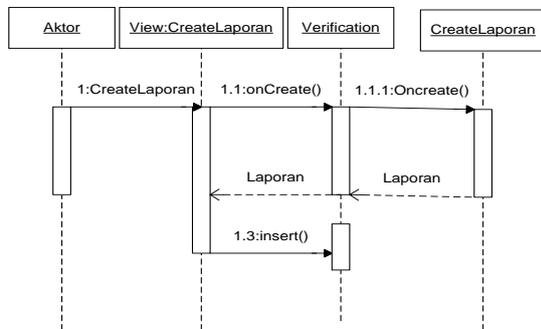


Gambar 24. Sequence Diagram Edit Profil

4. Sequence Diagram Create Laporan

Sequence digram *create laporan* menggambarkan tentang proses bisnis *create laporan* pada aplikasi *mobile*. pada

proses ini aktor memilih fungsionalitas *create* laporan kemudian aplikasi memunculkan *interface* create laporan setelah *interface* muncul, maka aktor diminta untuk memasukkan laporan barunya. kemudian sistem akan meverifikasi *inputan* tersebut, selanjutnya class CreateLaporan akan memberikan hasil eksekusi berupa laporan berhasil di *inputkan* atau tidak.



Gambar 25. Sequence Diagram Create Laporan

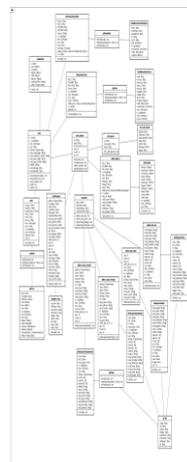
3. Class Diagram

Aplikasi sembakodroid memiliki 2 class diagram yang terdiri dari diagram aplikasi Android untuk petugas lapangan dan diagram aplikasi android untuk masyarakat. *Class diagram* aplikasi android untuk petugas lapangan digunakan untuk menggambarkan *class diagram* yang ada pada aplikasi sembakodroid untuk petugas lapangan. sedangkan *class digram* aplikasi android untuk msyarakat digunakan untuk menggambarkan *class diagram* pada

aplikasi sembakodroid yang diakses oleh masyarakat.

1. Class Diagram Aplikasi Android untuk Petugas Lapangan

Class digram aplikasi petugas pasar menggambarkan tentang *class digram* yang ada pada aplikasi sembakodroid untuk petugas lapangan.



Gambar 26. Class Diagram Aplikasi Android untuk Petugas Lapangan

2. Class Diagram Aplikasi Android Untuk Masyarakat

Class diagram aplikasi android untuk masyarakat menggambarkan tentang class diagram yang ada pada aplikasi sembako droid untuk masyarakat.



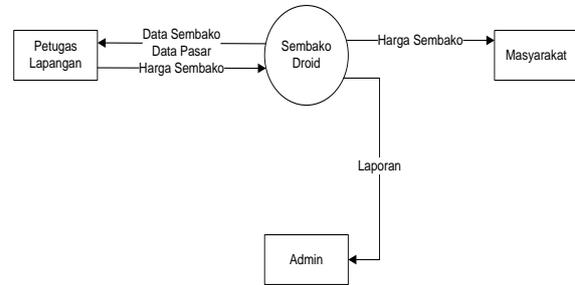
Gambar 27. Class Diagram pada Aplikasi Sembako Droid Di Android

DFD pada website

Aplikasi sembakodroid memiliki sebuah server yang berbasis website. dalam pembangunannya website ini dibangun berdasarkan bahasa pemrograman yang berbasis prosedural. oleh karena itu dibuatlah diagram DFD untuk menggambarkan proses tersebut. aplikasi sembako droid mempunyai 2 DFD yaitu DFD level 0 atau diagram konteks dan DFD level 1.

1. DFD level 0

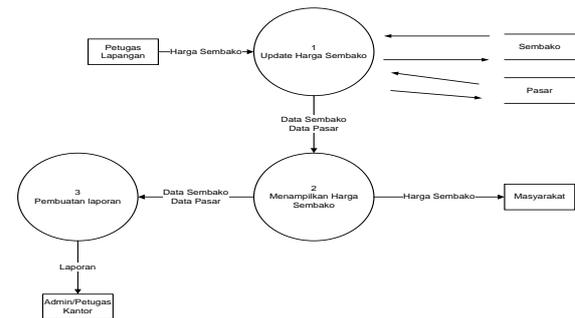
DFD level 0 menggambarkan proses bisnis pembangunan aplikasi pada aplikasi sembako droid yang berbasis web server.



Gambar 28. DFD level 0

2. DFD level 1

DFD level 1 merupakan turunan dari DFD level 0 yang menggambarkan detail dari proses bisnis aplikasi sembako droid yang berbasis web server



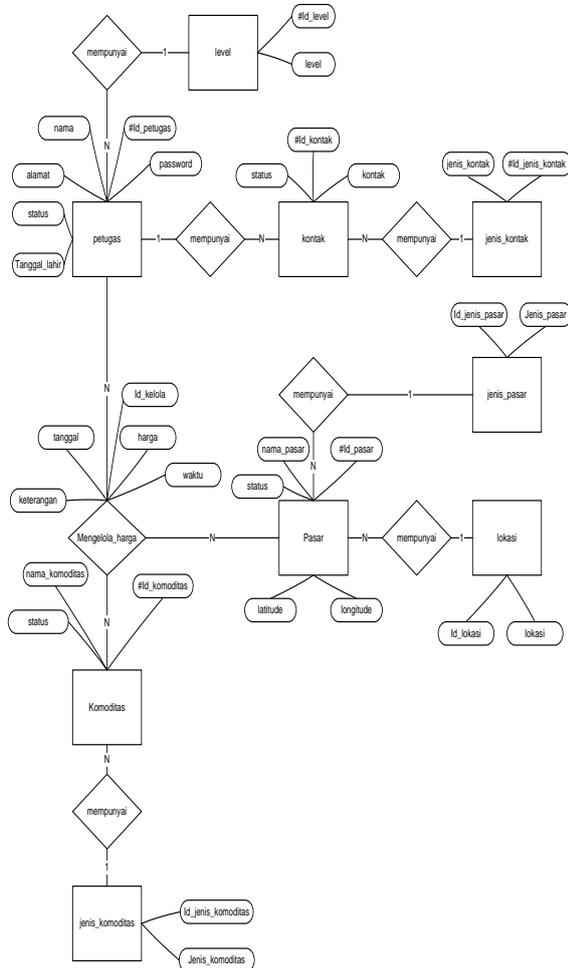
Gambar 29. DFD level 1

Perancangan Basis data

Setiap hari petugas pasar melakukan update terhadap data sembako di pasaran. data sembako tersebut membutuhkan sebuah database yang sangat dinamis karena data yang selalu berubah setiap hari.

1. ERD

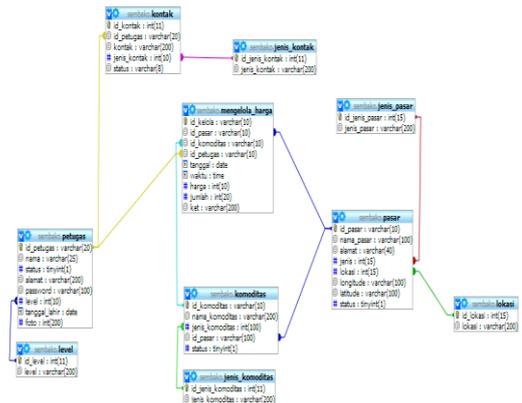
Perancangan ERD menggambarkan tentang gambaran database yang ada pada aplikasi sembakodroid



Gambar 30. Entity Relationship Diagram

2. Relasi Antar Tabel

Relasi antar tabel menggambarkan tentang hubungan antar tabel yang ada pada database aplikasi sembako droid



Gambar 31. Relasi Antar Tabel

3. Struktur Tabel

Menggambarkan struktur tabel pada Aplikasi sembakodroid monitoring harga sembako berbasis android antara lain.

a. Tabel Petugas

Primary key : id_petugas

Foreign key : level

Deskripsi : table petugas adalah tabel yang menampung data petugas. Yang terdiri dari beberapa atribut yaitu id_petugas, nama, alamat, dan password, level, tanggal_lahir, status.

Tabel 17. Struktur Tabel Petugas

NAME	TYPE	NOT NULL?	PRIMARY KEY
id_petugas	Varchar(20)	Yes	Yes
nama	Varchar(25)	Yes	No
alamat	Varchar(200)	No	No
password	Varchar(45)	No	No
level	Int(10)	No	No
tanggal_lahir	date	No	No
status	Tinyint(1)	No	No

b. Tabel Level

Primary key : id_level

Foreign key : -

Deskripsi : tabel level adalah tabel yang menampung data level petugas. Yang terdiri dari beberapa atribut yaitu id_level, level

Tabel 18. Struktur Tabel Level

NAME	TYPE	NOTNULL?	PRIMARY KEY
Id_level	int(11)	Yes	Yes
level	Varchar(200)	Yes	No

c. Tabel Kontak

Primary key : id_kontak

Foreign key : id_petugas,
jenis_kontak

Deskripsi : tabel kontak adalah tabel yang menampung data kontak petugas. Yang terdiri dari beberapa atribut yaitu id_kontak, id_petugas, kontak, jenis_kontak, status

Tabel 19. Struktur Tabel Kontak

NAME	TYPE	NOT NULL?	PRIMARY KEY
Id_kontak	int(11)	Yes	Yes
Id_petugas	Varchar(20)	Yes	No
kontak	Varchar(200)	Yes	No
jenis_kontak	Varchar(20)	Yes	No
Status	tinyint(1)	yes	No

d. Tabel Jenis_Kontak

Primary key : id_jenis_kontak

Foreign key : -

Deskripsi : tabel jenis_kontak adalah tabel yang menampung data jenis_kontak petugas. Yang terdiri dari beberapa atribut yaitu id_jenis_kontak, jenis_kontak.

Tabel 20. Struktur Tabel Jenis_Kontak

NAME	TYPE	NOTNULL ?	PRIMARY KEY
Id_jenis_kontak	int(11)	Yes	Yes
jenis_kontak	Varchar(200)	Yes	No

e. Tabel Pasar

Primary key : id_pasar

Foreign key : jenis, lokasi

Deskripsi : tabel pasar adalah tabel yang menampung data pasar. Yang terdiri dari beberapa atribut yaitu id_pasar, nama_pasar, alamat, jenis, lokasi, longitude, latitude, status

Tabel 21. Struktur Tabel Pasar

NAME	TYPE	NOT NULL?	PRIMARY KEY
Id_pasar	int(11)	Yes	Yes
nama_pasar	Varchar(100)	Yes	No
alamat	Varchar (40)	Yes	No
jenis	Int (15)	Yes	No
Lokasi	Int (15)	Yes	No
longitude	Varchar (100)	Yes	No
latitude	Varchar (100)	Yes	No
status	Tinyint (1)	Yes	No

f. Tabel jenis_pasar

Primary key : id_jenis_pasar

Foreign key : -

Deskripsi : tabel jenis_pasar adalah tabel yang menampung data jenis_pasar. Yang terdiri dari beberapa atribut yaitu id_jenis_pasar, jenis_pasar.

Tabel 22. Struktur Tabel Jenis_Pasar

NAME	TYPE	NOT NULL?	PRIMARY KEY
Id_jenis_pasar	int(15)	Yes	Yes
jenis_pasar	Varchar(200)	Yes	No

g. Tabel komoditas

Primary key : id_komoditas

Foreign key : jenis_komoditas

Deskripsi : tabel komoditas adalah tabel yang menampung data komoditas petugas. Yang terdiri dari beberapa atribut yaitu id_komoditas,

nama_komoditas,
jenis_komoditas, status.

Tabel 23. Struktur Tabel Komoditas

NAME	TYPE	NOT NULL?	PRIMARY KEY
Id_komoditas	int(15)	Yes	Yes
Nama_komoditas	Varchar(200)	Yes	No
jenis_komoditas	Int (100)	Yes	No
status	Tinyint(1)	Yes	No

h. Tabel jenis_komoditas

Primary key : id_jenis _komoditas

Foreign key : jenis_komoditas

Deskripsi : tabel jenis komoditas adalah tabel yang menampung data jenis komoditas. Yang terdiri dari beberapa atribut yaitu id_jenis _komoditas, jenis_komoditas.

Tabel 24. Struktur Tabel Jenis_Komoditas

NAME	TYPE	NOT NULL?	PRIMARY KEY
Id_jenis_komoditas	int(11)	Yes	Yes
jenis_komoditas	Varchar(200)	Yes	No

i. Tabel lokasi

Primary key : id_lokasi

Foreign key : -

Deskripsi : tabel lokasi adalah tabel yang menampung data lokasi. Yang terdiri dari beberapa atribut yaitu id_lokasi, lokasi.

Tabel 25. Struktur Tabel Lokasi

NAME	TYPE	NOT NULL?	PRIMARY KEY
Id_lokasi	int(15)	Yes	Yes
lokasi	Varchar(200)	Yes	No

j. Tabel mengelola_harga

Primary key : id_kelola

Foreign key : id_pasar, id_komoditas, id_petugas

Deskripsi : tabel mengelola _harga adalah tabel yang menampung data jenis_kontak petugas. Yang terdiri dari beberapa atribut yaitu id_kelola, id_pasar, id_petugas, id _komoditas, tanggal, harga, waktu, ket, jumlah.

Tabel 26. Struktur Tabel Mengelola_Harga

NAME	TYPE	NOT NULL?	PRIMARY KEY
Id_kelola	int(15)	Yes	Yes
Id_pasar	Varchar(10)	Yes	No
Id_petugas	Varchar(10)	Yes	No
Id_komoditas	Varchar(10)	Yes	No
tanggal	Date	Yes	No
harga	time	Yes	No
waktu	int(15)	Yes	No
ket	Varchar(200)	Yes	No
jumlah	int(20)	Yes	No

Kebutuhan Perangkat Keras dan Perangkat Lunak

1. Pengembangan Sistem

Untuk membangun aplikasi sembako droid dibutuhkan perangkat keras dan pernakat lunak dengan spesifikasi antara lain :

a. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam pembangunan aplikasi Sembakodroid antara lain:

Tabel 27. Kebutuhan Perangkat Keras Pengembangan Sistem

No.	Perangkat Keras	Keterangan
1.	Laptop / Notebook	Processor : Intel(R) Core(TM) 2 Duo Memory : 1 GB

No.	Perangkat Keras	Keterangan
2.	Gadged Android	RAM Harddisk : 160 GB Sistem Operasi : Android OS, v3.0 Chipset : Nvidia Tegra 2 T20 CPU : Dual-core 1 GHz Cortex- A9 Memory : 16/32 GB Storage

b. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam pembangunan aplikasi Sembakodroid antara lain:

Tabel 28. Kebutuhan Perangkat Lunak Pengembangan Sistem

No	Perangkat Lunak	Keterangan
Kebutuhan perangkat lunak pada aplikasi mobile		
1.	Eclipse Indigo Version 3.7.2	IDE aplikasi mobile
2.	Android SDK (<i>Software Development Kit</i>)	API(<i>Application Programming Interface</i>)
3.	Sun Java SE v1.6/v1.7	Compiler aplikasi android
4.	ADT (<i>Android Development Tools</i>)	Penghubung IDE Eclipse dengan Android SDK
5.	AVD (<i>Android Virtual Device</i>)	Emulator untuk menjalankan aplikasi android
Kebutuhan perangkat lunak pada aplikasi website		
6.	Operating System	Windows 7, Linux

No	Perangkat Lunak	Keterangan
7.	Mozilla Firefox/Chrome	Browser
8.	Adobe Dream Weaver	Design Aplikasi
9.	Notepad++	Editor bahasa pemrograman
10.	PhotoShop CS4	Editor gambar

Implementasi Sistem

Untuk implementasi aplikasi sembako droid dibutuhkan perangkat keras dan perngkat lunak dengan spesifikasi antara lain :

1. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam implementasi aplikasi Sembakodroid antara lain:

Tabel 29. Kebutuhan Perangkat Keras Implementasi Sistem

No.	Perangkat Keras	Keterangan
1.	Laptop / Notebook	Hp probook 4420 S Processor : Intel(R) Core i3 Memory : 1 GB RAM Harddisk : 160 GB
2.	Gadged Android	Sistem Operasi : Android OS, v 4.0.3 Chipset : Nvidia Tegra 2 T20 CPU : Dual-core 1 GHz Cortex- A9 Memory : 16/32 GB Storage Type : HTC EVO pro Design 4G Ukuran layar : 4 inch

2. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam pembangunan aplikasi Sembakodroid antara lain:

Tabel 30. Kebutuhan Perangkat Lunak Implementasi Sistem

No	Perangkat Lunak	Keterangan
Kebutuhan perangkat lunak pada aplikasi mobile		
1.	Eclipse Indigo Version 3.7.2	IDE aplikasi mobile
2.	Sun Java SE v1.6/v1.7	Compiler aplikasi android
4.	ADT (<i>Android Development Tools</i>)	Penghubung IDE Eclipse dengan Android SDK
Kebutuhan perangkat lunak pada aplikasi website		
6.	Operating System	Windows 7, Linux
7.	Mozilla Firefox/Chrome	Browser
8.	Adobe Dream Weaver	Design Aplikasi
9.	Notepad++	Editor bahasa pemrograman

HASIL DAN PEMBAHASAN

Pengujian

Pengujian *software* haruslah didefinisikan dalam proses rekayasa perangkat lunak atau *software engineering*. Sejumlah strategi pengujian software telah diusulkan dalam literatur. Semuanya menyediakan *developer software* dengan template untuk pengujian dan semuanya memiliki karakteristik umum:

1. Testing dimulai pada level modul dan bekerja keluar ke arah integrasi pada sistem berbasis computer.
2. Teknik testing yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya.
3. Testing diadakan oleh *software developer* dan untuk proyek yang besar oleh group testing yang independent.
4. *Testing* dan *Debugging* adalah aktivitas yang berbeda tetapi *debugging* harus diakomodasikan pada setiap strategi *testing*.

Pengujian *software* adalah satu elemen dari sebuah topik *broader* yang sering diartikan sebagai Verifikasi dan Validasi (V&V) :

- (1) Verifikasi : menunjuk ke kumpulan aktifitas yang memastikan bahwa *software* mengimplementasi sebuah fungsi spesifik.
- (2) Validasi : menunjuk ke sebuah kumpulan berbeda dari aktivitas yang memastikan bahwa *software* yang telah dibangun dapat di-trace terhadap kebutuhan customer.

Defenisi V&V meliputi banyak aktifitas SQA, termasuk review teknis formal, kualitas dan audit konfigurasi monitor performance, tipe yang berbeda dari pengujian *software study feasibility* dan simulasi. Metode *software engineering* menyediakan dasar dari mutu yang mana yang akan dipakai.

1. Metode *Analysis, design and Construction* berupa tindakan untuk meningkatkan kualitas dengan menyediakan teknik yang seragam dan hasil yang sesuai dengan keinginan.
2. Metode *Formal Technical Reviews* menolong untuk memastikan kualitas kerja produk merupakan hasil konsekuensi dari setiap langkah *software engineering*.
3. Metode *Measurement* diberlakukan pada setiap elemen dari konfigurasi *software*.
4. Metode *Standards and Procedures* membantu untuk memastikan keseragaman dan formalitas dari SQA untuk menguatkan dasar “filosofi kualitas total”.
5. Metode *Testing* menyediakan cara terakhir dari tingkat kualitas mana yang dapat dicapai dan dengan praktis dapat mengetahui letak error.

Adapun langkah-langkah pengujian *software* adalah sebagai berikut.

1. Unit *testing-testing* per unit yaitu mencoba alur yang spesifik pada struktur modul kontrol untuk memastikan pelengkapan secara penuh dan pendeteksian *error* secara maksimum.
2. *Integration testing-testing* per penggabungan unit yaitu pengalamatan dari isu-isu yang diasosiasikan dengan

masalah ganda pada verifikasi dan konstruksi program.

3. *High-order test* yaitu terjadi ketika *software* telah selesai diintegrasikan atau dibangun menjadi satu –tidak terpisah-pisah.
4. *Validation test* yaitu menyediakan jaminan akhir bahwa *software* memenuhi semua kebutuhan fungsional, kepribadian dan performa.

1. Kriteria dari *Testing* yang telah selesai

- a. Dengan menggunakan model statistik dan teori *software reliability*, model dari kegagalan *software* yang tidak terdeteksi selama *testing* sebagai fungsi dari waktu eksekusi dapat dikembangkan.
- b. Sebuah versi dari model kegagalan yang disebut *logarithmic Poisson execution-time model*.

Untuk mengimplementasikan *software testing strategy* yang sukses, di antaranya.

1. Menetapkan seluruh kebutuhan produk *software* dalam perhitungan sebelum memulai testing.
2. Status obyek *testing* harus jelas.

3. Memahami pengguna *software* dan mengembangkan sebuah profil untuk setiap kategori *user*.
4. Mengembangkan rencana *testing* yang menekankan pada “rapid cycle testing”.
5. Membangun *software* yang sempurna yang didesain untuk mengetes dirinya sendiri.
6. Menggunakan tinjauan ulang yang formal sebagai filter sebelum pengujian.
7. Melakukan tinjauan ulang secara formal untuk menilai strategi tes dan kasus tes itu sendiri.
8. Mengembangkan pendekatan peningkatan yang berkelanjutan untuk proses *testing*.

2. Test Unit (Test Level Komponen)

Pengujian unit: Komponen individual yang diuji secara independen untuk memastikan kualitasnya. Fokusnya untuk tidak menutup error pada desain dan implementasi, meliputi: struktur data pada sebuah komponen, logika program dan struktur program pada sebuah komponen, interface komponen, fungsi dan operasi dari sebuah component, dan adanya penguji/tester unit: pengembang dari komponen.

3. Test Integrasi

Test Integrasi merupakan sebuah group dari component dependent diuji bersama untuk memastikan kualitas dari

unit integrasinya. Merupakan teknik sistematis untuk membangun struktur program pada saat melakukan testing untuk mencari error. Secara obyektif untuk mengambil modul unit test dan membangun struktur program yang telah dirancang oleh desainnya. Fokusnya untuk meng-*uncover error* pada : Desain dan konstruksi arsitektur software, Fungsi-fungsi yang terintegrasi atau operasi pada level sub-system, Interface dan interaksi, Integrasi resource dan/atau integrasi lingkungan, dan Penguji integration: pengembang dan/atau *test engineer*.

Strategi Pengujian Integrasi melalui Pendekatan integrasi non-incremental melalui Big Band, menggabungkan (atau mengintegrasikan) semua bagian dalam sekali, dengan keuntungan: pengujian sangat sederhana dan kerugiannya adalah sulit untuk men-*debug*, tidak mudah untuk mengisolasi *error* tidak mudah untuk memvalidasi hasil test mustahil untuk membentuk sebuah sistem terintegrasi *impossible*. Sedangkan integrasi *incremental* dengan mengintegrasikan sistem tahap demi tahap (atau bagian demi bagian) dalam sebuah pesanan yang didesain dengan baik. Tiga metode penting, di antaranya : *Top-down*, *bottom-up*, dan *Sandwich*.

4. Integrasi Top-down

Modul-modul diintegrasikan dengan memindahkan downward melalui struktur kontrol. Modul subordinate ke modul kontrol utama digabung ke sistem dalam cara *depth-first* atau *breadth-first*. Proses integrasi (lima langkah) :

- a. Modul kontrol utama digunakan sebagai sebuah test driver, dan stubs disubstitusi untuk semua modul secara langsung ke modul kontrol utama.
- b. Stub subordinate digantikan sekali satu waktu dengan modul actual.
- c. Test terkonduksi sebagai tiap modul diintegrasikan.
- d. Pada pelengkapan tiap kumpulan test, stub lainnya diganti dengan modul real.
- e. Pengujian regresi dapat dikonduksi.

Integration top-down pros dan cons : biaya konstruksi stub dan fungsi kontrol utama dapat diuji lebih cepat.

5. Integrasi Bottom-Up

Modul pada level terbawah diintegrasikan pertama, kemudian dengan menggerakkan keatas melalui struktur kontrol. Proses integration (lima langkah) :

- a. Modul low-level dikombinasikan ke cluster yang menunjukkan

sebuah sub-function software spesifik.

- b. Sebuah driver ditulis untuk meng-coordinate input dan output test case.
- c. Test cluster diuji.
- d. Driver dipindah dan cluster digabungkan bergerak ke atas dalam struktur program.

6. Integrasi bottom-up pros dan cons: tidak ada biaya stub, perlu pengujian driver, dan tidak ada sistem yang dapat dikontrol hingga langkah terakhir.

7. Regression Testing

Merupakan aktivitas yang membantu untuk memastikan sebuah perubahan (yang berkaitan dengan testing atau penjelasan lain) tidak menghasilkan perilaku yang tidak diharapkan atau error tambahan. Regression tes terdiri dari 3 kelas, yaitu:

- a. Sebuah Contoh yang mewakili tes yang akan menguji semua fungsi software.
- b. Tes tambahan yang berfokus pada fungsi software yang tampak yang akan berubah akibat perubahan lain.
- c. Tes yang berfokus pada komponen software yang telah berubah.
- d. Validation testing. Uji Validasi : Software yang berintegrasikan diuji

berdasarkan pada kebutuhan untuk memastikan bahwa kita memiliki produk yang benar. Fokus-nya adalah untuk meng-uncover error pada: Input/output sistem, Informasi fungsi sistem dan data, Interface sistem dengan bagian eksternal, User interface, Perilaku dan performance sistem.

8. Uji Sistem

Uji sistem: Sistem software diuji keseluruhan. Ini memverifikasi semua elemen secara langsung untuk memastikan bahwa semua fungsi dan performance sistem diterima dalam lingkungan target. Terbagi menjadi 5 bagian yaitu :

- a. Recovery Testing : sistem tes yang menekan software untuk gagal dengan cara yang bervariasi dan memverifikasi perbaikan sendiri dengan baik.
- b. Security Testing : usaha untuk memverifikasi mekanisme perlindungan yang dibuat dalam sistem apakah akan melindunginya dengan semestinya.
- c. Stress Testing : didesain untuk menghadapi program dengan situasi abnormal.
- d. Performance Testing : didesain untuk menguji performa software

ketika bekerja dalam konteks pengintegraian sistem.

- e. Pengujian instalasi : didesain untuk menguji prosedur instalasi dan software pendukungnya.

9. Rencana Uji

Rencana uji berhubungan dengan mengeset standar untuk pengujian proses dibanding penggambaran pengujian produk. *Test plan*/rencana uji terdiri atas : standar untuk proses pengujian, *resource* yang diperlukan (*hardware, software* dan *engineer*), jadwal pengujian (pengujian *task* dan *milestones*), uji item (apa yang harus diuji), prosedur *recording test* (hasil tes harus secara sistematis direkam), dan constraint. *Test planning* adalah sebuah dari suatu *test manager*. Sebuah *test plan* adalah *output* dari perencanaan.

10. Debugging

Gejalanya mungkin sebagai hasil dari masalah pemilihan waktu dibanding masalah proses. Kemungkinan sulit secara akurat mereproduksi kondisi input. Gejalanya mungkin terjadi dalam waktu yang dekat berkelanjutan. Gejalanya mungkin berkaitan dengan penyebab yang didistribusikan melewati sejumlah

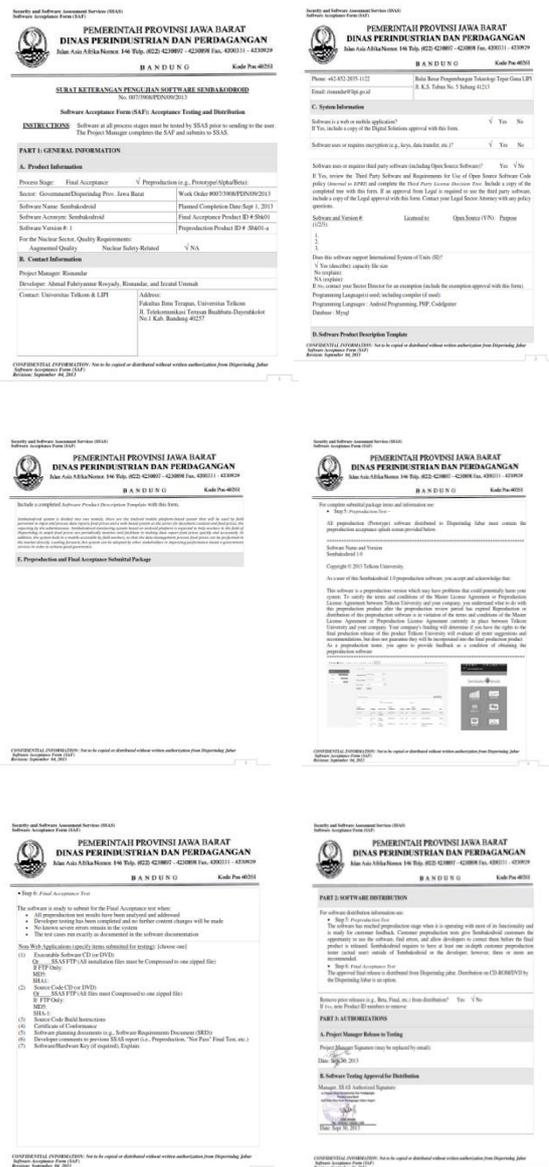
task yang berjalan pada prosesor yang berbeda. Terdapat 3 kategori pendekatan Debugging :

- Brute force : metode yang paling umum dan lebih efisien untuk mengisolasi penyebab dari error software.
- Backtracking : metode yang dapat berhasil pada program kecil.
- Cause elimination : perwujudan dari induksi atau deduksi dan pengenalan dari konsep binary partitioning.

Diskusi dan Pembahasan

Pengujian aplikasi sembako droid dilakukan dengan menggunakan black box testing. *black box testing* merupakan sebuah pengujian yang dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsionalitas dari *interface* perangkat lunak. pada pengujian black box testing, pengujian dilakukan oleh *developer* selaku pembuat sistem dan dikomunikasikan dengan *user* sebagai pemohon kebutuhan *user*.

Berikut disampaikan bukti Software Acceptance Form (SAF): Acceptance Testing and Distribution yang dilakukan oleh Dinas Perindustrian dan Perdagangan terhadap Sistem Sembakodroid.



Gambar 32. Relasi Antar Tabel

KESIMPULAN

Berdasarkan penelitian ini, maka dapat disimpulkan :

- Sistem ini dapat menentukan pusat perdagangan komoditas sembako tertentu dengan sistem yang terkomputerisasi dengan berbasis mobile
- Sistem ini dapat menentukan suatu daerah yang mempunyai potensi

penjualan paling banyak terhadap suatu komoditas tertentu

- 3) Sistem ini dapat membuat laporan harga sembako secara periodik

Sedangkan pengembangan sistem ke depannya lebih diarahkan kepada :

- 1) Membuat interface yang lebih menarik dan inovatif.
- 2) Menambahkan fungsionalitas yang dirasa masih kurang dalam sistem ini.
- 3) Memperbaiki proses pengelolaan dan pembuatan laporan yang dirasa masih kurang dalam sistem ini.

UCAPAN TERIMA KASIH

Penelitian ini dapat terlaksana atas dukungan dari para peneliti dan staf Balai Besar Pengembangan Teknologi Tepat Guna-LIPI, Dosen di Program Studi Manajemen Informatika, Fakultas Ilmu Terapan dan Program Studi Ilmu Komputasi, Fakultas Teknik, Universitas Telkom.

DAFTAR PUSTAKA

- Ali Zaky, E. W. (2012). *Membuat Web Mobile Dengan JQuery Mobile*. Jakarta: Elex Media Komputindo.
- Arbie. (2004). *Manajemen Database dengan MySQL*. Yogyakarta: Andi.
- Borom, E. (2000). *Study Offers Early Look at How Internet is Changing Daily Life*. Stanford Institute for the Quantitative Study of Society.
- Broulik, B. (2011). *Pro jQuery Mobile*. Minnesota: Apress.
- Doe, J. (2000). *Internet Usage Within Nations*. Boston: Boston Publishing.
- Internet World Stats. (2006). *Top 10 Countries With The Highest Numbr of Internet Users*. Diakses dari Internet World Stats: sage and Population Statistics: <http://www.internetworldstats.com/top20.htm> pada 12 30, 2006
- Interprise, J. (2011). *Step by step html*. Jogjakarta: Elex Media Komputindo.
- Irmansyah, F. (2003). *Pengantar Database*.
- Jayadi, M. (2010, Januari 11). *Implementasi ISO 9001:2008 di Perguruan Tinggi*. Bandung.
- Kadir, A. (2008). *Dasar Pemograman Web Dinamis Menggunakan PHP*. Yogyakarta: Andi.
- Lubis, J. (2001). *Internet User Behaviour*. McMillan Publishing.
- M Huda, K. (2011). *Membuat Aplikasi Database dengan Java, MySQL dan NetBeans*. Jakarta: Elexmedia Media Komputindo.
- Mellers. (2000). Choice and the relative pleasure of consequences. *Psychological Bulletin* , 5.
- Moenir. (1992). *Manajemen Pelayanan Umum*. Jakarta: Bumi Aksara.
- Nugroho, E. P. (2009). *Rekayasa Perangkat Lunak*. Bandung: Politeknik Telkom.
- Roberts, S. (2009). *Information System: Now and Tomorrow*. Chicago: Adventure Press.
- Rohim, A. (2009). *Perancangan Basis Data Relasional*. Bandung: Politeknik Telkom.

- Rohit Ghatol, Y. P. (2012). *Beginning PhoneGap Mobile Web Framework for JavaScript and HTML 5*. Switzerland: Apress.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak*. Andi Publisher.
- Safaat, N. (2011). *ANDROID*. Bandung: Penerbit Informatika.
- Saputro, T. (2011). *Aplikasi dan Arsitektur Android*. Surabaya: Putra Jatim.
- Soeherman, B. D. (2008). *Designing Information System*. Jakarta: Elex Media Komputindo.
- Sommerville, I. (2007). *Software Engineering*. Addison Wesley.
- Sunyoto, A. (2007). *AJAX-Asynchronous JavaScript & XML*. Yogyakarta: Penerbit Andi.
- Supardi, D. (2006). *Sistem Kerja Perpustakaan Daerah (15 ed.)*. Jakarta: Gramedia.
- Suryatiningsih, W. M. (2009). *Pemrograman WEB*. Bandung: Politeknik Telkom.
- Sutanta. (2004). *Sistem Basis Data*. Yogyakarta: Graha Ilmu.
- Syafaat, N. (2011). *Pemrograman Mobile Smartphone Dan Tablet PC Berbasis Android*. Bandung: Elex Media Komputindo.
- Wijaya, D. (2009). *Perancangan Basis Data*. Bandung: Politeknik Telkom.