Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

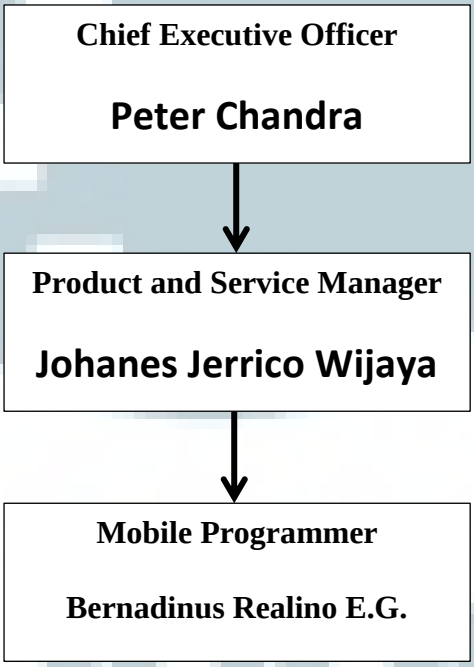
Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama bekerja bersama dengan peserta yang lain pada minggu pertama diberikan tugas yang sama yaitu sebagai *Web Designer*, kemudian pada minggu berikutnya baru mulai ditugaskan sebagai *Mobile Programmer*. Kerja magang ini diawasi oleh Bapak Johanes Jerrico Wijaya sebagai *senior programmer* dan juga sebagai manajer *Product and Service*.



Gambar 3.1 Koordinasi Kerja Magang

Rancang bangun..., Bernadinus Realino Edy Gunawan, FTI UMN 2014

3.2 Tugas yang Dilakukan

Modul kiosk memiliki beberapa modul kecil yang lain seperti *Layouting*, *Slideshow*, *Slideshow Timer Controll*, *Dynamic Content*.

Pada *Layouting*, ditugas untuk membuat *page* sesuai dengan desain yang telah dibuat oleh desainer dan telah disetujui oleh manajer. Cara pengerjaannya, dengan menggunakan *view* untuk yang berperan sebagai *container*, pada *view* tersebut nantinya akan diisi dengan *button*, *imageView*, *scrollView*, *scrollableView*, *label* yang masing – masing digunakan sesuai dengan kebutuhan.

Slideshow dan *Slideshow Timer Controll* yang akan digunakan dengan sebagai *slideshow*, yang mana semua data yang nanti akan digunakan akan disediakan dalam *Dynamic Content*.

Dikarenakan data yang diambil harus bersifat fleksibel yang mana vendor dapat menentukan apa yang ingin ditampilkan. Oleh karena itu database yang digunakan langsung diubah dan menyesuaikan dengan databasenya, dengan memanfaatkan *Dynamic Content* yang sudah disediakan oleh peserta magang yang lain.

System Development Life Cycle menjadi acuan untuk merealisasikan aplikasi *Kiosk Mobile Loyalti Express*. Adapun pengembangan aplikasi yang akan dikembangkan ini dimulai dari tiga fase awal yaitu *initiation*, *system concept development*, dan *planning*. Dimulai dengan berkoordinasi dengan Bapak Johanes Jerrico selaku *project manager* agar dapat mengetahui bagaimana proses yang harus dikerjakan. Pada tahap selanjutnya, mulai lebih banyak memberikan masukan pada pembuatan sistem dari rancangan yang sudah ada, karena untuk perencanaan dan perancangan awal tidak terlalu dipublikasikan untuk meminimalisir keluarnya informasi yang bersifat internal.

Setelah tiga persiapan awal sudah dilaksanakan mulai diberikan tanggung jawab untuk melanjutkan fase – fase berikutnya dalam pembuatan sistem *Kiosk Mobile Loyalti Express* berikut keterangan *timetable* dari fase - fase tersebut yang dapat dilihat pada table 3.1.

Table 3.1 Timeline Pelaksanan Proyek

No	Kegiatan	Minggu ke-							
		1	2	3	4	5	6	7	8
1	*Studi literature*								
2	*Requirement analysis*								
3	*Design*								
4	*Development*								
5	*Integration and test*								
6	*Implementation*								
7	*Maintenance*								
8	Penulisan dokumentasi								

3.2.1 Studi Literatur

Dalam studi literature dimulai dengan melakukan pembelajaran dengan dokumentasi yang ada pada jurnal *online* mengenai titanium studio agar dapat lebih mendalami cara menggunakanya dan juga mencoba untuk membuat sebuah uji coba suatu fungsi agar dapat lebih memahami secara langsung prose penggunaannya.

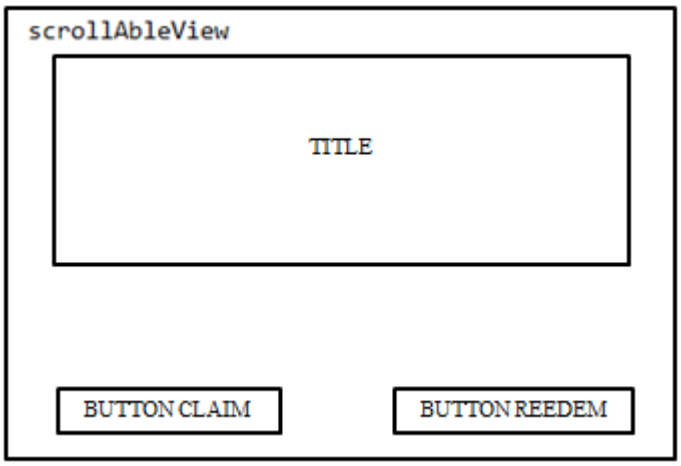
3.2.2 Requirement Analysis

Kiosk adalah sistem antar muka yang menyediakan informasi dengan metode elektronik, yang pada umumnya ditempatkan pada tempat umum. Oleh karena itu untuk dapat memberikan informasi yang akurat dan menarik untuk dilihat oleh customer desain yang diberikan dan penyampaian informasinya haruslah jelas.

A. Modul Layouting

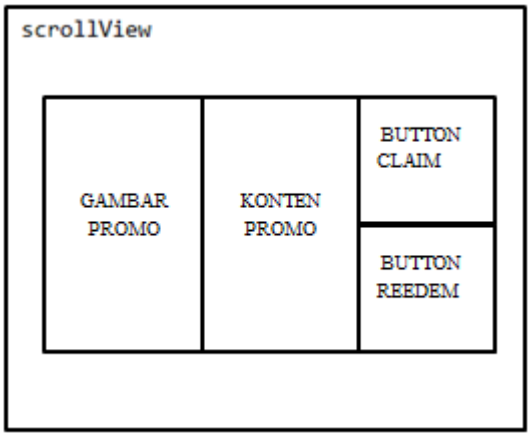
Pada modul *Layouting* pertama mulai menganalisa *tool* yang akan digunakan untuk membuat desain – desain. *Tool* yang digunakan sebagai berikut :

1. *Template slideshow* terdiri dari *scrollAbleView* yang befungsi agar dapat melakukan *scroll*, lalu membuat sebuah *view* yang bernama *title* yang memuat isi kontent dari *slideshow*, dan dua tombol *button* yaitu *claim* dan *redeem*,untuk *button redeem* tidak dibuat modulnya.



Gambar 3.2 Sketsa Slideshow

2. *Template* desain utama pertama terdiri dari *scrollVIew* yang berfungsi agar dapat melakukan scroll secara perlahan atau dapat diatur, lalu *imageView* yang digunakan untuk menampilkan gambar promo. Label yang digunakan untuk menampilkan konten promo dan dua *button* yaitu *claim* dan *redeem*, untuk *button redeem* tidak dibuat modulnya

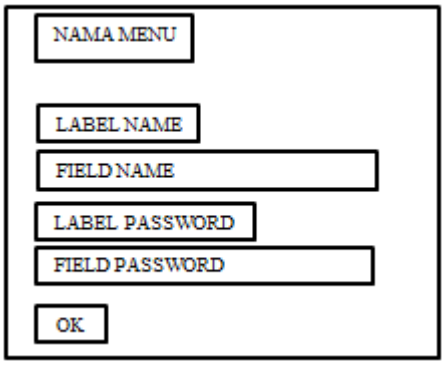


Gambar 3.3 Sketsa Desain Utama Pertama

3. Template desain utama kedua ini terdiri dari *scrollAbleView* yang befungsi agar dapat melakukan *scroll*, dan pada perhalamannya di siapkan tiga buah konten yang masing – masing memiliki promo yang berbeda dan juga memiliki tiga buah deskripsi yang menyesuaikan dengan masing – masing konten. Konten terdiri dari *view* dan deskripsi dari *label*.

Gambar 3.4 Sketsa Desain Utama Kedua

4. Template login terdiri dari tiga label yaitu untuk nama menu, *name*, dan *password*, dua buah *fieldText* yang digunakan untuk menampung *name* dan *password* dari *user*, dan sebuah *button* ok yang digunakan untuk menyelesaikan proses tersebut.



Gambar 3.5 Sketsa Login

B. Modul Slideshow

Modul *Slideshow* di sini diharapkan untuk membuat *Slideshow* yang terlihat dinamis, lancar dan dapat menampilkan data sesuai yang dibutuhkan, untuk menyelesaikan masalah tersebut dengan menggunakan *scrollableView* karena mendukung kontrol paging pada layar untuk menunjukkan apakah halaman sebelumnya atau berikutnya ada, lalu pada setiap halamannya tersebut mulai membuat *view* yang diberikan fungsi pengulangan dan dibuat menjadi array agar dapat menampung isi dari *view* tersebut yang kesemuanya dibuat diatas

imageView yang akan menjadi *background* nya tersebut, *imageView* tersebut juga dibuat menjadi array yang sudah diatur isinya berupa gambar dari *background* yang berubah apabila customer sudah melakukan *scroll*.

C. Modul Slideshow Timer Controll

Modul *Slideshow Timer Controll* disini, diharuskan membuat sebuah tool yang mana ini digunakan untuk mengatur perpindahan antara halaman – halaman yang sudah diakses langsung kembali lagi menuju halaman *Slideshow*, cara kerjanya yaitu membuat suatu penampung agar dapat mengetahui halaman mana saja yang sudah diakses dan disimpan dengan nama *pageView* yang berupa array. Lalu sebuah *timer* yang digunakan sebagai penanda berapa lama transisi akan terjadi. Setelah waktu yang ditentukan sudah berjalan maka *pageView* yang ada akan ditutup karena data yang ada didalam *pageView* tersebut merupakan alamat dari halaman atau sebuah *windows* kemudian dikeluarkan satu persatu agar *pageView* menjadi kosong lalu langsung mengarah ke halaman *Slideshow*.

D. Modul Dynamic Content

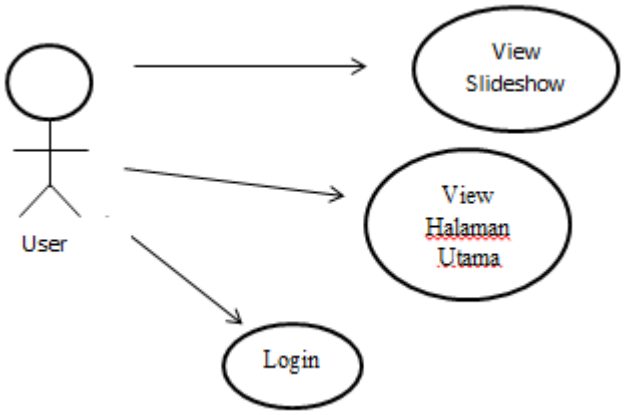
Untuk modul ke empat yaitu Dynamic Content, disini mulai menggunakan *tool* yang dibuat oleh Joshua Prayogi Angki yang bernama *JSON to Backbone JS Mapper*. *Tool* ini berfungsi untuk membuat isi dari aplikasi secara *dynamic*. Cara kerjanya *tool* ini adalah mengambil data dari *server* yang berupa *JSON String* yang kemudian akan diubah menjadi *JSON Object*.

3.2.3 Design

Setiap sistem pasti memiliki rancangan yang menggambarkan bagaimana sistem tersebut berjalan. Untuk dapat lebih menjelaskan rancangan teknis tentang alur kerja dari modul yang sudah dibuat.

Berikut ini adalah beberapa gambaran sistem yang dirancang dalam bentuk diagram yang nantinya akan di implementasikan secara langsung menjadi sistem.

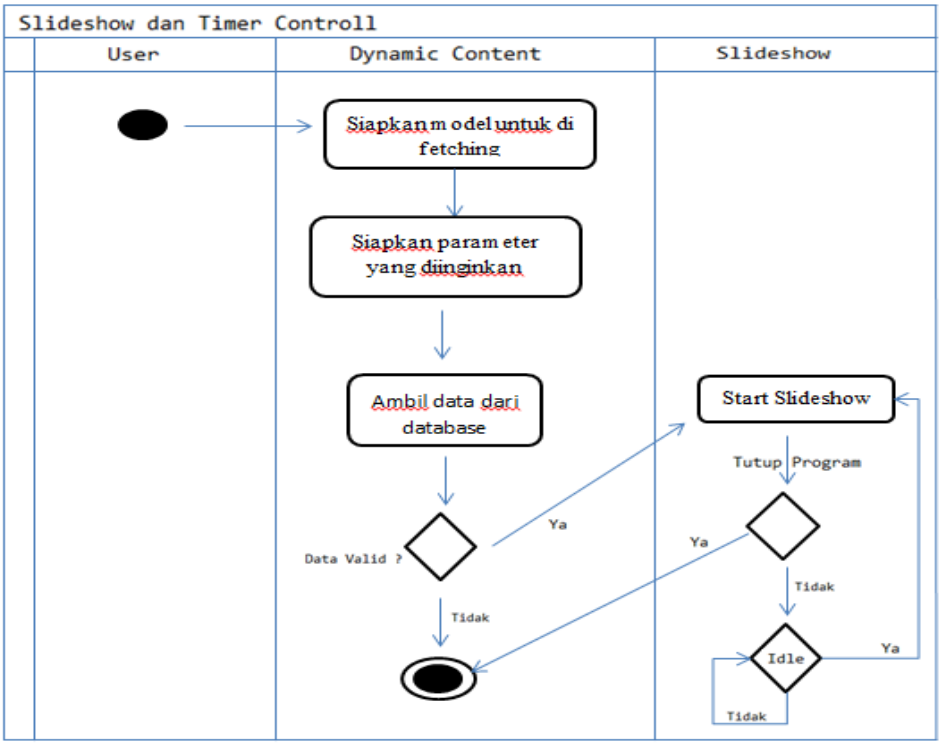
a. Use Case Diagram



Gambar 3.6 *Use Case Diagram*

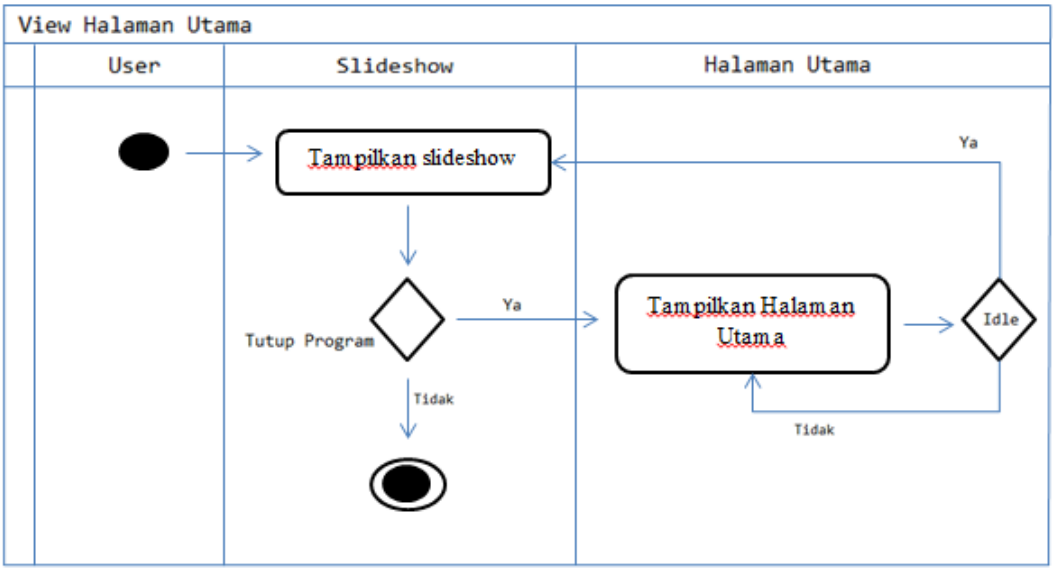
Proses kerja utama yang ada adalah melakukan *preview* untuk *slideshow* dan juga halaman utama dan user dapat melakukan login namun modul untuk login bukan merupakan bagian dari modul yang dikerjakan.

b. Activity Diagram



Gambar 3.7 *Activity Diagram Slideshow* dan *Timer Controll*

Activity diagram pada gambar 3.7 ini menjelaskan bagaimana proses dari *Slideshow* dimulai dari penyiapan database yang nanti akan digunakan, kemudian mulai menjalankan *Slideshow Timer Controll* nya.



Gambar 3.8 Activity Diagram View Halaman Utama

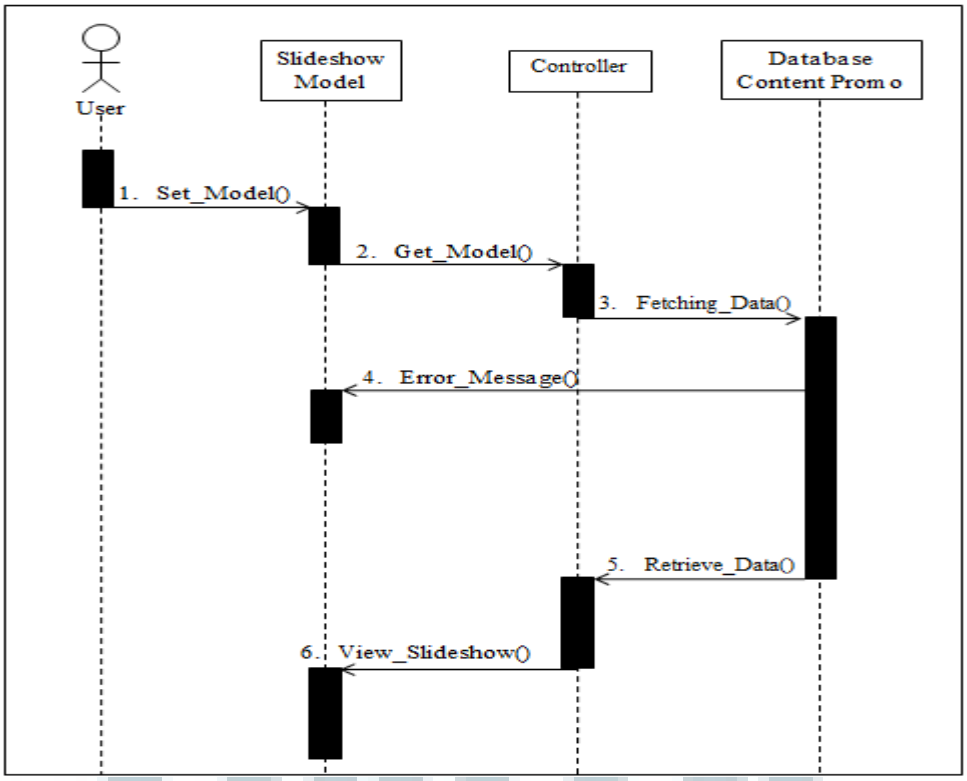
Activity diagram pada gambar 3.8 menggambarkan bagaimana proses untuk melihat halaman utama yang menampung *promo* yang ada.

c. Sequential Diagram

1. *Sequence Diagram Slideshow*

Skenario sebagai berikut.

1. User menyiapkan model yang akan digunakan untuk mengambil data
2. Model yang telah disiapkan akan digunakan pada controller sebagai penentu data apa saja yang akan diambil
3. Mulai melakukan penarikan data pada database
4. Apabila terjadi kesalahan pada penarikan data maka akan dikembalikan pesan error
5. Data telah diterima
6. Setelah data diterima maka data tersebut akan digunakan untuk membuat isi dari kontent slideshow



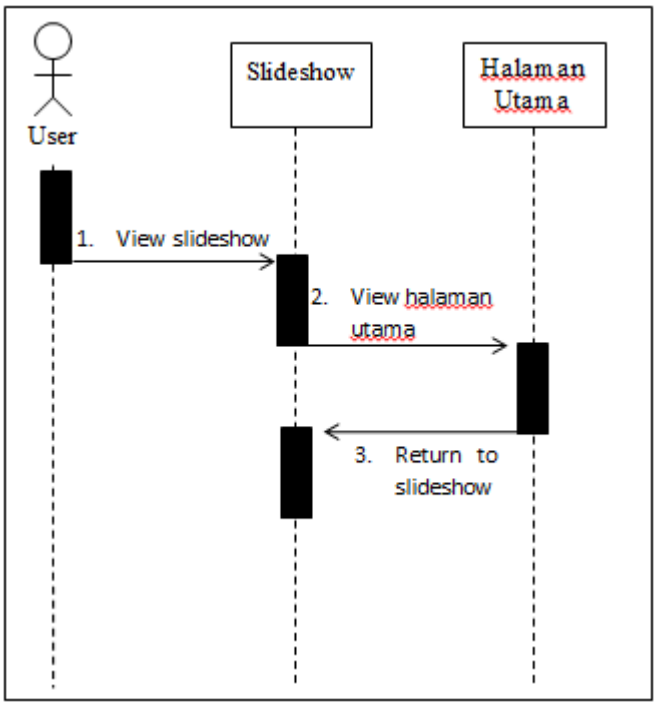
Gambar 3.9 *Sequential Diagram Slideshow*

Gambar 3.9 menggambarkan *sequential diagram* yang menjelaskan bagaimana proses dari *Slideshow Timer Controll* dimulai dari penyiapan database yang nanti akan digunakan, untuk menjalankan slideshow.

2. *Sequence Diagram View* Halaman Utama

Skenario sebagai berikut.

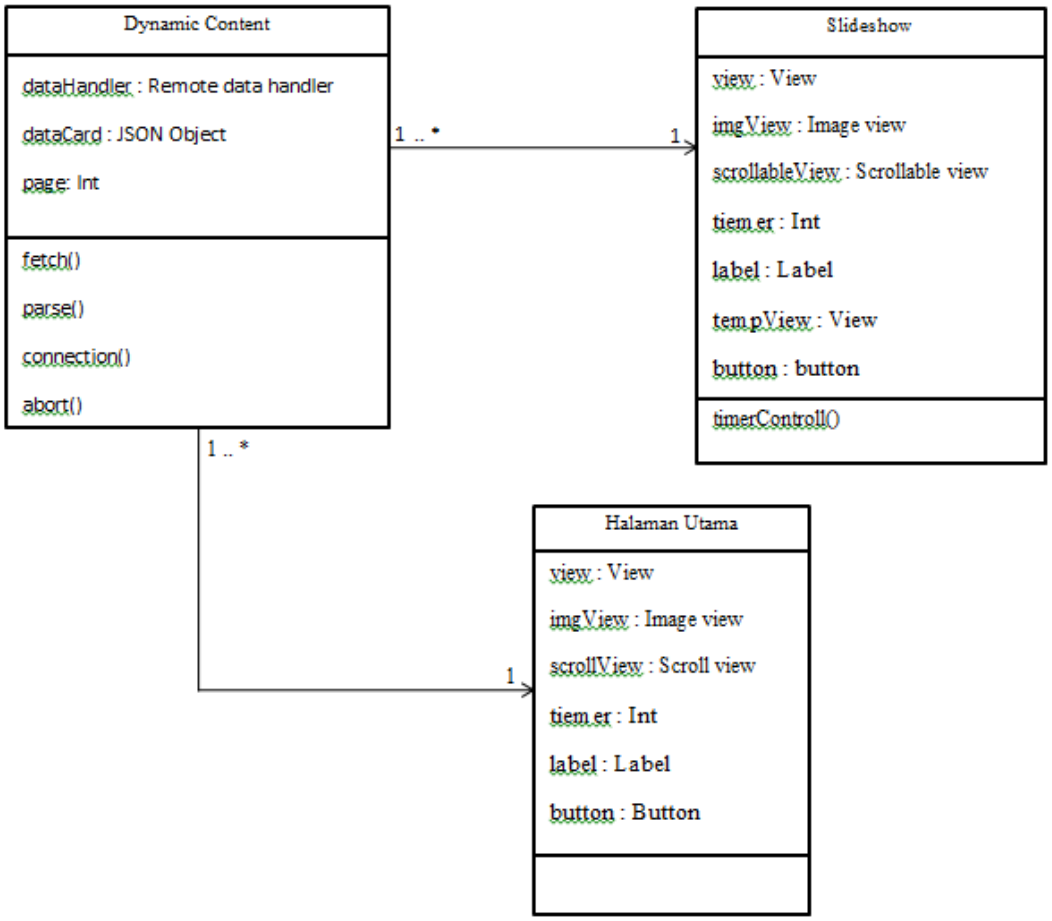
1. User masuk ke *slideshow*
2. User masuk ke halaman utama untuk dapat melihat – lihat promo yang ada
3. Apabila user sudah tidak melihat – melihat promo lagi, maka setalah lewat dari waktu yang ditentukan maka akan kembali lagi ke *slideshow*



Gambar 3.10 *Sequential Diagram View* Halaman Utama

Gambar 3.10 menggambarkan *sequence diagram* untuk dapat melihat promo – promo yang sedang ada sekarang.

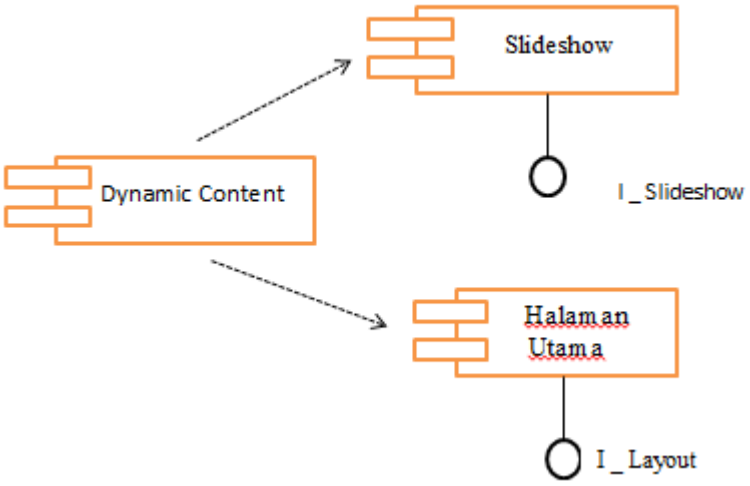
d. Class Diagram



Gambar 3.11 Class Diagram

Gambar 3.11 menjelaskan hubungan setiap *class* yang ada di dalam sistem. Dari *class diagram* tersebut dapat dilihat bahwa satu buah *slideshow* dan halaman utama memiliki satu atau lebih *dynamic content*.

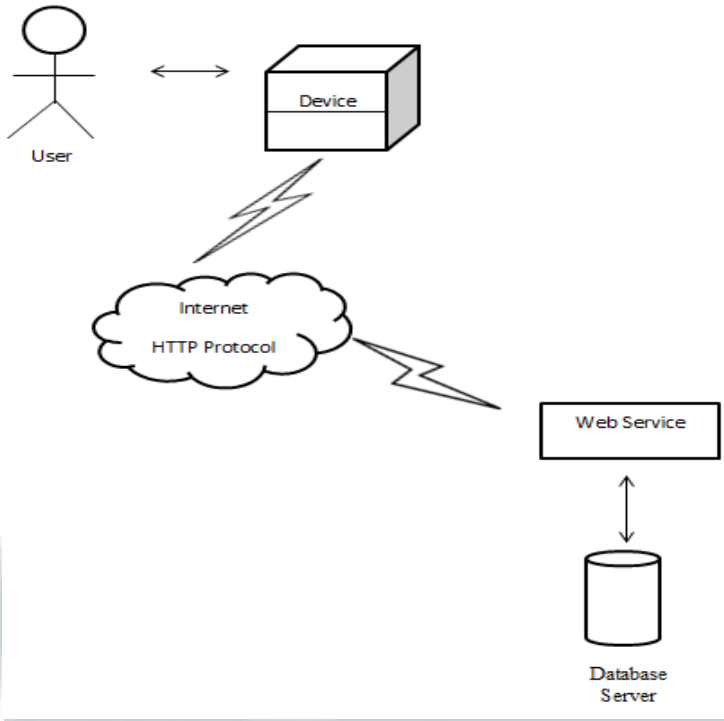
e. Component Diagram



Gambar 3.12 *Component Diagram*

Component diagram ini menggambarkan hubungan – hubungan antar komponen yang ada pada suatu sistem tersebut.

f. Deployment Diagram



Gambar 3.13 *Deployment Diagram*

Diagram ini menjelaskan bagaimana proses sistem yang digunakan oleh program tersebut dan hubungan - hubungannya.

3.2.4 Development

Selama proses pengembangan dari sistem ini, ada beberapa perangkat keras dan perangkat lunak yang akan digunakan. Berikut ini adalah penjabaran perangkat keras yang digunaka.

1. Laptop Asus A46CB sebagai perangkat yang digunakan selama pengembangan dengan spesifikasi umum sebagai berikut.
   a. Processor Intel Core i5-3217U (1.70 GHz, Cache 3MB)
   b. VGA NVIDIA GeForce GT740M 2GB
   c. RAM dengan kapasistas 4Gb
   d. Harddisk dengan kapasitas 500Gb
2. Tablet Asus Google Nexus 7 sebagai perangkat untuk melakukan *testing* dengan spesifikasi sebagai berikut.
   a. Processor Quad-core 1.2 GHz Cortex-A9
   b. Chipset Nvidia Tegra 3
   c. Operating System Android v4.2.2 (Jelly Bean)
   d. RAM 1Gb
   e. Internal Memory dengan kapasistas 32Gb
3. Perangkat input yaitu *mouse* dan *keyboard*
4. Komputer *development server* and *storage*

Beberapa perangkat lunak yang digunakan selama tahap pengembangan sistem antara lain sebagai berikut.

1. Sistem operasi Windows 8 untuk laptop.
2. Appcelerator Titanium Studio IDE, digunakan sebagai coding editor berbasis JavaScript untuk mengembangkan Multi-platform mobile application.
3. Android Software Development Kit untuk dapat menjalankan fitur – fitur android pada emulator.
4. Android Virtual Emulator, digunakan untuk menjalankan simulasi dari aplikasi yang dibuat.
5. Google Chrome web browser yang digunakan untuk mencari informasi perihal pengembangan sistem.

6. Microsoft Office Excel 2010 untuk membuat timesheet dan laporan pertanggung jawaban kerja.
7. Microsoft Office Word 2010 untuk membuat dokumentasi dan laporan kerja magang.

3.2.5 Integration and Test

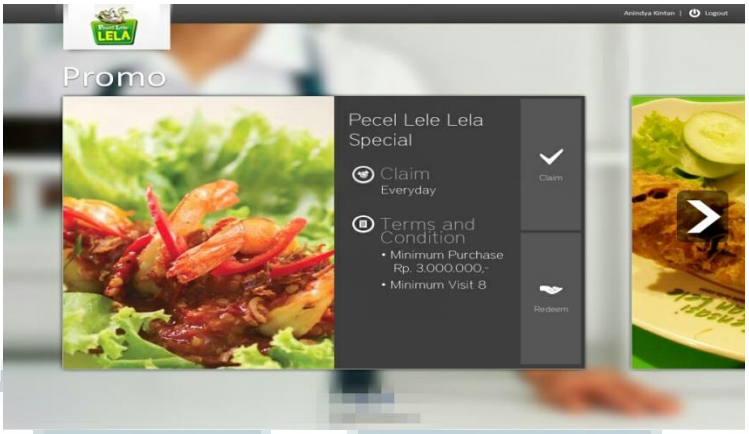
Selama pengembangan sistem ada beberapa pengujian yang dilakukan untuk mengetahui apakah aplikasi yang dibuat dapat memenuhi kriteria yang sudah disediakan. Point – point yang diperhatikan dalam pengujian seperti apakah data yang diambil untuk *layout* sudah benar, kemudian melakukan pengujian *memory leak* agar tidak sering terjadi *crash*, setelah itu dilakukan pengujian terhadap ke sensitivitas aplikasi tersebut.

Pada point pertama hanya ditekankan pada, apakah data yang ditampilkan sesuai dengan yang diharapkan apabila sudah sesuai makan akan dilanjutkan kepada point yang berikutnya.

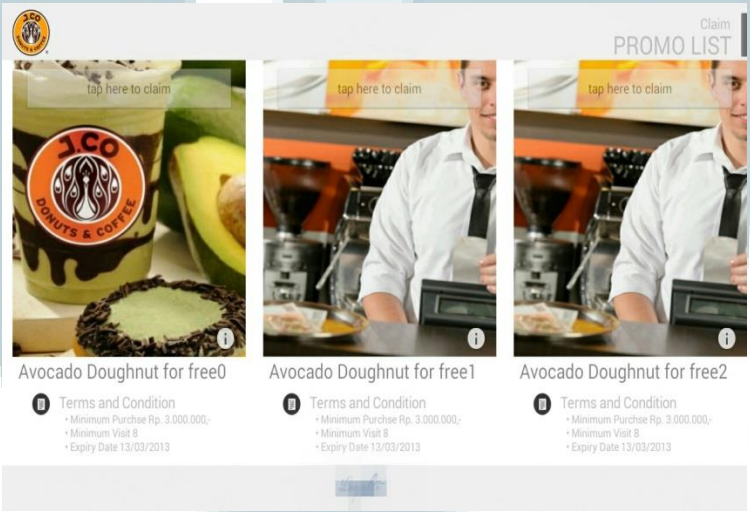
Kemudian point berikutnya, mulai melakukan pengujian dengan cara membiarkan aplikasi nya menyala dalam waktu yang lama, kemudian menutup aplikasi tersebut dan membukanya kembali beberapa kali, yang terakhir memberikan gerakan yang cepat untuk mencoba *scrollableView* pada *tablet* yang digunakan lalu mencoba menekan bagian *button* yang disediakan dan melihat reaksi yang diberikan oleh aplikasi apabila aplikasi dapat memberikan respon dengan baik dan benar maka aplikasi tersebut sudah berhasil dibuat.

3.2.6 Implementation

Dalam proses implementation ini pertama dimulai dengan membuat *layout*. Desain utama pertama menggunakan *scrollableView* namun karena kurang sesuai dengan kriteria *project manager* maka dirubah menjadi *scrollView*. Namun kemudian desain tersebut diubah menjadi yang kedua, untuk itu kembali menggunakan *scrollableView* karena seperti yang sudah dijelaskan diatas, yaitu *scrollableView* mendukung kontrol paging pada layar untuk menunjukkan apakah halaman sebelumnya atau berikutnya ada.

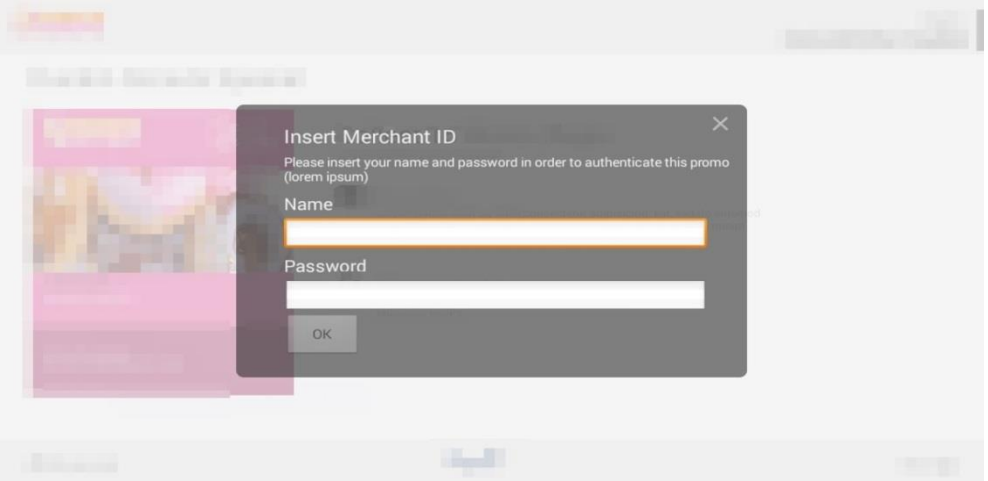


Gambar 3.14 Desain Utama Pertama



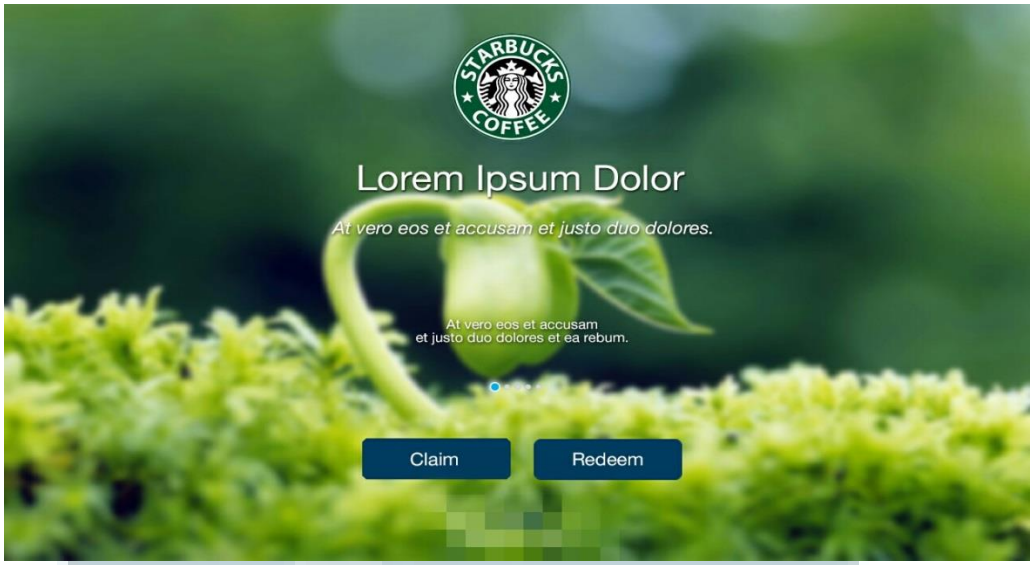
Gambar 3.15 Desain Utama Kedua

Kemudian mulai membuat desain yang lain seperti desain untuk *login* nya, untuk desain *login* desain dibuat agar terlihat dinamis dan tidak membuang tempat pada saat merchant ingin login. Dengan mengatur opacity desain menjadi terlihat menarik dan juga terlihat menggunakan animasi *fade in – fade out.*



Gambar 3.16 Login

Berikutnya membuat *Slideshow*, disini kesulitan yang ditemui ada pada pembuatan desain agar bergerak mengikuti dengan *scroll* yang dilakukan. Namun setelah melakukan beberapa percobaan akhirnya ditemukan juga cara untuk mengerjakannya yaitu dengan cara menggunakan *scrollableView* dengan membuat *view* didalamnya. Pada *view* tersebut baru dibuat desain sesuai yang diinginkan seperti menambahkan *label*, *button* dan *imageView* setelah semuanya selesai maka pada saat *scroll* mulai digerakan gambar image yang ada dibelakangnya akan mengikuti gerakan *scroll* nya lalu yang terakhir menambahkan efek *fade-in* dan *fade-out* pada saat melakukan *scroll* dan setelah selesai *scroll*.



Gambar 3.17 *Slideshow*

Pada *Dynamic Content* hanya menggunakan function yang telah dibuat oleh Joshua Prayogi Angki yang memiliki fungsi sebagai *tool* untuk mengambil data dari *server* yang berupa *JSON String* yang kemudian akan diubah menjadi *JSON Object. JSON Object* yang digunakan pada saat ingin menggunakan gambar yang sudah disiapkan agar sesuai dengan data yang telah disiapkan jadi tidak perlu mengambil atau menyiapkan gambar pada *data local* namun hanya perlu menakses pada *server* yang telah disiapkan.

Kemudian untuk membuat *Slideshow Timer Controll* seperti yang telah dijelaskan diatas function ini digunakan untuk mengarahkan suatu halaman tertentu kembali lagi ke halaman slideshow. Pada pembuatan function ini hal yang paling menjadi kendala adalah Pada saat menyimpan dan mengeluarkan alamat halaman yang dibutuhkan sering terjadi kesalahan pada saat penyimpanan dan pengeluarannya terutama saat dilakukan pengulangan pada function tersebut dan juga pada saat menghapus *event timer* nya agar tidak menyala lagi pada saat berpindah halaman. Namun untuk dapat menemukan solusi untuk itu yaitu membuat *variable* ynag menyimpan halaman tersebut kedalam sebuah *array* lalu pada saat melakukan perpindahan, alamat dari halaman tersebut akan di *pop* atau dimasukan kedalam *array* lalu pada saat timer sudah berjalan maka alamat yang ada pada *array* akan langsung di *push* semua atau dikeluarkan lalu langkah yang

paling penting adalah pada setiap terjadi perpindahan halaman *eventListener* yang digunakan menjadi *timer* harus di *remove* kemudian *timer* yang sudah disiapkan di *clearTimeOut* dengan begini maka *timer* yang sudah ada tidak akan saling bertabrakan dengan *timer* yang baru pada halaman yang baru.

3.2.7 Maintenance

Untuk *maintenance*, diharuskan membuat aplikasi tersebut menjadi fleksible jadi tidak memiliki ketergantungan data antara setiap halaman yang ada, mempermudah saat *debugging* untuk mencari *error* dan juga mempermudah pada saat dikerjakan oleh *programmer* lain untuk dikembangkan.

3.2.8 Penulisan dokumentasi dan laporan magang

Setiap hari semua diharuskan untuk memberikan perincian kerja harian yang nanti digunakan sebagai *control* dari kantor untuk mengukur kinerja dari peserta magang. Untuk dokumentasi aplikasi, dengan membuat *comment* yang bisa dibaca didalam program untuk memperjelas setiap bagian – bagian yang *program* yang ada.

Untuk memberikan gambaran yang lebih jelas, berikut ini tampilan rincian kerja harian selama berkerja di PT Moonlay Technologies dalam bentuk table.

Table 3.2 Realisasi Kerja Magang

Minggu	Kegiatan
1	- Pengenalan dan pembentukan team - Membuat *layout* promo untuk website
2	- Optimasi *layout* promo untuk website - Membuat fungsi *Knockout JS*
3	- Optimasi *layout* promo untuk website - Pengenalan tentang Appcelerator Titanium Studio
4	- Membuat *layout* untuk Loyalti Express - Optimasi *layout* untuk Loyalti Express
5	- Membuat *layout* untuk Loyalti Express - Membuat aplikasi menjadi fleksibel - Membuat *Prototype Dynamic Content*
6	- Membuat *layout* untuk Loyalti Express - Optimasi *layout* untuk Loyalti Express
7	- Membuat *Slideshow* untuk Loyalti Express - Optimasi *layout* untuk Loyalti Express - Membuat *function* Slideshow Timer Controll
8	- Implementasi *Jason Mapper* kedalam modul *Dynamic Content* - Revisi dan penyelesaian modul - modul

3.3 Kendala yang Ditemukan

Untuk membuat sebuah aplikasi yang baik dan dapat memenuhi kebutuhan customer, tentunya akan ada banyak kendala yang ditemukan. Dalam pengerjaan magang ini pun menemukan kendala, baik itu kendala teknis maupun non teknis, berikut ini beberapa kendala yang ditemukan selama pengerjaan.

3.3.1 Kendala Teknis

Beberapa kendala teknis yang ditemukan seperti dalam modul – modul yang dikerjakan sebagai berikut.

a. Penggunaan Appcelerator Titanium Studio, karena baru pertama kali digunakan dalam mengembangkan aplikasi. Untuk dapat membangun aplikasi tersebut tentunya dibutuhkan keahlian yang cukup dalam penguasaan bahasa pemograman tersebut. Kendala waktu pengembangan yang cukup lama karena dibarengi dengan studi literatur untuk membangun aplikasi yang sesuai dengan kebutuhan.

b. Dalam menyesuaikan layout dengan desain yang sudah ditentukan. Dari posisi gambar, posisi tulisan, dan posisi tombol yang ada disana.

c. Mekanisme untuk menggunakan *Slideshow timer*. Banyak terjadi kendala pada saat mengatur perpindahan dari halaman ke *slideshow* nya.

d. Aplikasi harus dapat berjalan dengan cepat dan lancar, oleh karena itu harus pemilihan algoritma yang tepat merupakan cara terbaik untuk dapat mencapai kecepatan dan kelancaran yag diinginkan.

3.3.2 Kendala Non Teknis

Beberapa kendala non teknis yang ditemui sebagai berikut.

a. Kebutuhan dari sistem yang sering berubah – ubah yang membuat proses pengerjaan jadi berjalan lebih lambat dari waktu yang sudah diperkirakan.

b. Keterbatasan *resource* yang juga menghambat proses pengerjaan. Untuk membangun sebuah *mobile application* tentunya dibutuhkan *device* yang dapat digunakan untuk melakukan testing yang tidak disediakan oleh pihak kantor.

3.4 Solusi atas Kendala yang Ditemukan

Setiap kendala pasti memiliki solusi, untuk menyelesaikan kendala yang didapat maka didapatkan solusi sebagai berikut ini.

3.4.1 Solusi untuk Kendala Teknis

a. Melakukan studi literatur dan pembelajaran terhadap Titanium Studio. Ditambah juga mencari contoh - contoh program yang bisa menjadi bekal dalam menyelesaikan tugas yang ada.

b. Setelah sudah belajar dan mencoba berkali – kali, dengan modal pengalaman yang sudah didapat setelah menyelesaikan layout – layout yang ada, pembuatan desain menjadi lebih ringan.

c. Solusi yang didapat adalah mulai dengan meperhatikan setiap *eventListener* yang ada, meperhatikan apakah *eventListener* sudah di *remove* atau masih aktif. Dengan begitu tidak perlu khawatir dengan *eventListener* yang akan terbawa, apabila terjadi perpindahan.

d. Setelah melakukan beberapa riset dan percobaan seperti mengganti ukuran dari *image* yang dibutuhkan, mengurangi jumlah *button*, *image*, dan *label* yang digunakan. Membuat desain menjadi lebih terorganisir dengan menggunakan *view*.

3.4.2 Solusi untuk Kendala Non Teknis

Beberapa solusi yang ditemukan menyelesaikan beberapa kendala teknis dalam pengembangan sistem adalah sebagai berikut.

a. Seringnya terjadi perubahan selama proses pembuatan, dapat diatasi dengan menggunakan *layout* yang sudah dibuat sebelumnya sebagai panduan untutk membuat *layout* yang baru karena biasanya desain yang diberikan tidak terlalu berbeda jauh. Lalu diadakan rapat harian untuk pembahasan kerja untuk menghindari salah koordinasi dalam pengerjaan. Hal ini dimaksudkan juga agar deadline untuk setiap proyek yang dikerjakan tidak mundur karena adanya kesalahan dasar dalam pengembangan.

b. Untuk mengatasi keterbatasan *resource*, menggunakan *device* yang sudah ada untuk melakukan percobaannya.