

Android PoS Cloud-based

TUGAS AKHIR

Oleh :

Yohan Nor Cahyo P 3311211030

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2015**

HALAMAN PENGESAHAN

Android PoS Cloud-based

Oleh :

Yohan Nor Cahyo P (3311211030)

Tugas Akhir ini telah diterima dan disahkan
sebagai persyaratan untuk memperoleh gelar

Ahli Madya

di

**PROGRAM STUDI DIPLOMA 3 TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM**

Batam, 29 Januari 2015

Disetujui oleh;

Pembimbing I,

Pembimbing II,

Andy Triwinarko, MT
NIK. 105038

Yeni Rokhayati, M.Sc
NIK. 112093

HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : 3311211030

Nama : Yohan Nor Cahyo P

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

Android PoS Cloud-based

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 29 Januari 2015

Yohan Nor Cahyo P
3311211030

KATA PENGANTAR

Alhamdulillah segala puji syukur penulis panjatkan kehadiran Tuhan, karena atas limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul “Android PoS Cloud-based” sesuai waktu yang disediakan. Tugas Akhir ini disusun sebagai persyaratan kelulusan Program Diploma III pada program studi Teknik Informatika Politeknik Negeri Batam.

Dalam penyusunan Tugas Akhir ini penulis banyak mendapatkan saran, dorongan, bimbingan serta keterangan-keterangan dari berbagai pihak yang merupakan pengalaman yang tidak dapat diukur secara materi, namun dapat membuka mata penulis bahwa sesungguhnya pengalaman dan pengetahuan tersebut adalah guru terbaik bagi penulis. Maka dari itu dengan segala hormat dan kerendahan hati perkenankanlah penulis mengucapkan terima kasih kepada :

1. Tuhan Yang Maha Esa, Berkat rahmat yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini tepat waktu.
2. Bapak Andy Triwinarko, MT selaku Pembimbing I.
3. Bapak Tandy Bintang S, S.Kom selaku pemberi ide.
4. Bapak Dwi Eli Kurniawan, MT selaku Koordinator TA.
5. Ibu Yeni Rokhayati, Msc selaku Pembimbing II.
6. Bapak Syahrial selaku *user* requirement pembuatan TA ini.
7. Orang Tua yang memberikan dukungan secara moril.
8. Teman-teman kelas IFA tahun 2012, yang selalu mmberikan motivasi.

Dalam penulisan Tugas Akhir ini, penulis menyadari masih terdapat banyak kekurangan yang di buat baik sengaja maupun tidak sengaja, dikarenakan keterbatasan ilmu pengetahuan dan wawasan serta pengalaman yang penulis miliki. Untuk itu penulis mohon maaf atas segala kekurangan tersebut, tidak menutup diri terhadap segala saran dan kritik serta masukan yang bersifat konstruktif bagi diri penulis.

Akhir kata semoga dapat bermanfaat bagi penulis sendiri, institusi pendidikan dan masyarakat luas.

Batam, Januari 2015

Penulis

ABSTRAK

Android PoS Cloud-based

Aplikasi Poin of sale (PoS) adalah aplikasi yang digunakan untuk membantu proses transaksi dalam bidang penjualan. Aplikasi ini banyak dikembangkan pada desktop atau komputer, ada juga sebagian yang sudah menggunakan *mobile smartphone*. Namun aplikasi-aplikasi tersebut masih bersifat *local database* atau media penyimpanan masih didalam *hardisk* komputer tersebut. Hal ini sangat berbahaya jika suatu saat data-data transaksi tersebut hilang akibat kerusakan *hardware*. Sedangkan pengguna atau pemilik usaha ingin melihat atau mengambil data yang terdahulu guna kepentingan laporan. *Cloud* adalah sebuah teknologi penyimpanan data yang berada disuatu tempat yang tidak diketahui oleh pemakai. Keuntungan menggunakan cloud adalah pengguna tidak perlu membuat *server* dengan biaya yang sangat mahal serta melakukan perawatan *server* dengan biaya yang mahal juga. Dengan adanya *cloud* semua data dapat terintegrasi dan dapat diakses dari manapun dengan syarat harus terkoneksi dengan internet. Dari keadaan diatas jika mengkorelasikan dengan perkembangan teknologi di bidang *Smartphone* dan *cloud computing* maka tercipta gagasan baru. *Smartphone* yang semakin canggih dimungkinkan dapat mendukung kegiatan PoS tersebut. Sedangkan media penyimpanan data dapat di *back up* dengan teknologi *cloud (SaaS)* sehingga dapat mengurangi kehilangan data yang disebabkan kerusakan *hardware*. Selain itu pemilik usaha juga dapat melihat perkembangan usahanya dimana pun berada dengan syarat harus terkoneksi dengan internet. Aplikasi ini berhasil dibangun dengan memiliki kemampuan untuk menjalankan proses pemesanan secara mandiri oleh pembeli, proses pembayaran, pengecekan hasil penjualan, pembuatan *user* baru, dan pembuatan menu atau harga baru.

Kata kunci : *Android, Cloud Computing, Point of Sale (PoS)*

ABSTRACT

Android PoS Cloud-based

Application Point of Sale (PoS) is the application that is used to help the process transaction in selling. It's many developed for desktop, computer, and also smartphone mobile. But that applications are still shaped as database local or storage media in the computer hard disk. This matter is quite dangerous if the transaction data's lost due to the damage hardware. Meanwhile, the *user* or business owner would like to see or proceed the previous data for report purposes. Cloud is data storage technology which is located somewhere unknown by the *user*. The advantage using cloud is the *user* no need to create server with the high price and also doing treatment of it. With the existence of cloud, all the data could be integrated and accessed from any locations with terms which must be connected with the internet. From the above condition, if correlate with technology development in smartphone and cloud computing then it's created a new idea. More sophisticated of smartphone, it may be supported the PoS activities. Meanwhile, data storage media could be backed up by Cloud technology (SaaS) then it can reduce lost data that is caused by the damage hardware. Moreover, the business owner could also see the business development wherever they are, the terms is which must be connected with the internet. This application succeed built with having the ability to run the process of reserving autonomously by buyers , the payment process , checking the sale , creating a new *user* , and creating a menu or the new price.

Key Words : *Android, Cloud Computing, Point of Sale (PoS)*

DAFTAR ISI

HALAMAN PENGESAHAN	i
HALAMAN PERNYATAAN.....	ii
KATA PENGANTAR.....	iii
ABSTRAK	iv
ABSTRACT	v
DAFTAR ISI	vi
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Sistematika Penulisan.....	2
BAB 2 LANDASAN TEORI	3
2.1 Pengertian PoS	3
2.2 Pengertian Android	3
2.2.1 Android SDK (Software Development Kit)	4
2.2.2 ADT (Android Development Tools)	4
2.3 Pengertian Cloud	4
2.4 Arsitektur Cloud Computing.....	5
2.5 Mekanisme Cloud Computing	5
2.5.1 Software as a Service (SaaS)	5
2.5.2 Platform as a Service (PaaS)	6
2.5.3 Infrastructure as a Service (IaaS)	6
2.6 Pengertian Eclipse	6
2.7 Pengertian Java.....	7
2.8 Pengertian UML	7
2.9 Pengertian PHP	9
2.10 Pengertian MySQL.....	9
2.11 Pengertian JSON	10
BAB 3 ANALISA DAN PERANCANGAN	13

3.1 Analisa.....	13
3.1.1 Analisa Masalah	13
3.1.2 Analisa Kinerja Sistem Secara Umum	13
3.2 Analisa Kebutuhan Sistem	13
3.2.1 Analisa Kebutuhan Hardware.....	14
3.2.2 Analisa Kebutuhan Software	14
3.2.3 Analisa Kebutuhan SDM(Sumber Daya Manusia)	14
3.2.4 Analisa Kebutuhan Fungsional.....	14
3.4 Diskripsi Umum Sistem	15
3.5 Diagram Use Case Android PoS Cloud-based.....	16
3.6 Definisi Aktor.....	16
3.7 Definisi Use Case	17
3.8 Skenario Use Case.....	17
3.8.1 Nama Use Case : <i>login</i>	17
3.8.2 Nama Use Case :Kelola Pembayaran	18
3.8.3 Nama Use Case :Kelola Pesanan.....	18
3.8.4 Nama Use Case :Kelola <i>User</i>	19
3.8.5 Nama Use Case :Kelola Harga	19
3.8.5 Nama Use Case :Laporan	20
3.9 Sequence Diagram Android PoS Cloud-based.....	21
3.9.1 Sequence Diagram <i>login</i>	21
3.9.2 Sequence Diagram Kelola Pembayaran	22
3.9.3 Sequence Diagram Kelola Pesanan	22
3.9.5 Sequence Diagram Keola <i>User</i>	23
3.9.6 Sequence Diagram Kelola Harga	23
3.9.6 Sequence Diagram Laporan.....	24
3.10 Class Diagram Android PoS Cloud-based	24
3.11 Perancangan Basis Data	25
3.12 Perancangan Interface Android PoS Cloud-based	25
3.12.1 Antar Muka <i>login</i>	25
3.12.2 Antar Muka Home Kasir	26
3.12.2 Antar Muka Home Vendor	27
3.12.3 Antar Muka Pembayaran	27
3.12.4 Antar Muka Kelola Harga	28

3.12.5 Antar Muka Buat Harga	29
3.12.6 Antar Muka Lihat Harga.....	29
3.12.7 Antar Muka Kelola <i>user</i>	30
3.12.8 Antar Muka Buat <i>User</i>	31
3.12.9 Antar Muka Lihat <i>User</i>	32
3.12.9 Antar Muka Laporan	33
3.12.10 Antar Muka Pesanan.....	34
BAB 4 HASIL DAN PEMBAHASAN	35
4.1 Prosedur Aplikasi Android PoS Cloud-based	35
4.2 Perancangan Aplikasi Android PoS Cloud-based.....	35
4.3 Impementasi Database.....	35
4.4 Impementasi Antar Muka (Interface)	37
4.4.1 Impementasi Frame <i>login</i>	37
4.4.2 Impementasi Frame Home Vendor.....	39
4.4.3 Impementasi Frame Daftar Menu	40
4.4.4 Impementasi Frame Pesanan	40
4.4.5 Impementasi Frame Laporan Vendor	42
4.4.6 Impementasi Button Keluar.....	43
4.4.7 Impementasi Frame Home Kasir	43
4.4.8 Impementasi Frame Pembayaran.....	45
4.4.9 Impementasi Kelola <i>User</i> (Daftar <i>User</i>).	46
4.4.10 Impementasi Kelola <i>User</i> (Buat <i>User</i>).	46
4.4.11 Impementasi Kelola <i>User</i> (Edit dan Delete <i>User</i>).....	47
4.4.12 Impementasi Kelola Harga (Daftar Harga).	48
4.4.13 Impementasi Kelola Harga (Buat Harga).	49
4.4.14 Impementasi Kelola Harga (Edit dan Delete Harga).....	50
4.4.15 Impementasi Laporan Kasir.....	51
4.5 Impementasi Cloud	53
4.6 Pengujian Android PoS Cloud-based.....	53
4.6.1 Tujuan Pengujian.....	53
4.6.2 Strategi Pengujian.....	53
4.6.3 Hasil Pengujian.....	53
BAB 5 KESIMPULAN DAN SARAN	61
5.1 Kesimpulan.....	61

5.2 Saran.....	61
DAFTAR PUSTAKA.....	62
LAMPIRAN	63

DAFTAR GAMBAR

Gambar 1. Struktur <i>Cloud Computing</i>	5
Gambar 2. Objek dalam JSON	11
Gambar 3. Array dalam JSON.....	11
Gambar 4. Nilai dalam JSON.....	11
Gambar 5. String dalam JSON	12
Gambar 6. Angka dalam JSON	12
Gambar 7. Diskripsi Umum Sistem.....	15
Gambar 8. Use Case Android PoS Cloud.....	16
Gambar 9. Sequence Diagram <i>login</i>	21
Gambar 10. Sequence Diagram Kelola Pembayaran.....	22
Gambar 11. Sequence Diagram Kelola <i>User</i>	23
Gambar 12. Sequence Diagram Kelola Harga.....	23
Gambar 13. Sequence Diagram Laporan.....	24
Gambar 14. Class Diagram Android PoS Cloud-based.....	24
Gambar 15. Perancangn Basis Data	25
Gambar 16. Antar Muka <i>login</i>	25
Gambar 17. Antar Muka Menu Kasir	26
Gambar 18. Antar Muka Home Vendor	27
Gambar 19. Antar Muka Pembayaran	27
Gambar 20. Antar Muka Kelola Harga	28
Gambar 21. Antar Muka Buat Harga.....	29
Gambar 22. Antar Muka Lihat Harga.....	29
Gambar 23. Antar Muka Kelola <i>User</i>	30
Gambar 24. Antar Muka Buat <i>User</i>	31
Gambar 25. Antar Muka Lihat <i>User</i>	32
Gambar 26. Antar Muka Laporan.....	33
Gambar 27. Antar Muka Pesanan.....	34
Gambar 28. Frame <i>login</i>	38
Gambar 29. Frame Home Vendor	39
Gambar 300. Daftar Menu.....	40
Gambar 311. Frame Pesanan	41

Gambar 322. Frame Laporan Vendor.....	42
Gambar 333. Button Keluar	43
Gambar 344. Frame Home Kasir.....	44
Gambar 355. Frame Pembayaran	45
Gambar 366. Frame Kelola <i>User</i> (Daftar <i>User</i>)	46
Gambar 377. Frame Kelola <i>User</i> (Buat <i>User</i>).....	47
Gambar 388. Frame Kelola <i>User</i> (Edit & Delete <i>User</i>).....	48
Gambar 390. Kelola Harga(Buat Harga).....	50
Gambar 401. Kelola Harga(Edit Dan Delete Harga).....	51
Gambar 412. Laporan Kasir	52

DAFTAR TABEL

Table 1. Hardware Komputer	14
Table 2. Kebutuha Software Pengembangan.....	14
Table 3. Diskripsi Aktor	16
Table 4. Definisi Use Case Android PoS Cloud-based	17
Table 5. Skenario Use Case <i>login</i>	17
Table 6. Use Case Pembayaran	18
Table 7. Use Case Pesanan	18
Table 8. Use Case Kelola <i>User</i>	19
Table 9. Use Case Kelola Harga.....	20
Table 10. Use Case Laporan.....	21
Table 11. Deskripsi Antar Muka <i>login</i>	26
Table 12. Deskripsi Antar Muka Menu kasir	26
Table 13. Deskripsi Antar Muka Vendor	27
Table 14. Deskripsi Antar Muka Pembayaran.....	28
Table 15. Deskripsi Antar Muka Kelola Harga.....	28
Table 16. Deskripsi Antar Muka Buat Harga	29
Table 17. Deskripsi Antar Muka Lihat Harga	30
Table 18. Deskripsi Antar Muka Kelola <i>User</i>	30
Table 19. Deskripsi Antar Muka Buat <i>User</i>	31
Table 20. Deskripsi Antar Muka Lihat <i>User</i>	32
Table 21. Deskripsi Antar Muka Laporan	33
Table 22. Deskripsi Antar Muka Pesanan	34
Table 23. Tabel File PHP	53
Table 24. Tabel Pengujian	54

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dikutip dari Wikipedia, *Cloud Computing* adalah hasil dari evolusi bertahap dimana sebelumnya terjadi fenomena *grid computing*, *virtualisasi*, *application service provision(ASP)*, dan *Software as a Service (SaaS)*. Konsep penyatuan *computing resource* melalui jaringan global dimulai pada tahun 60-an. Salah satu batu loncatan yang cukup drastis adalah dengan adanya *Salesforce.com* ditahun 1999, yang merupakan pencetus pertama aplikasi dijalankan melalui *internet*. Perkembangan berikutnya adalah *Amazon Web Service* di tahun 2006, dimana dengan teknologi *Elastic Compute Cloud(EC2)*, terdapat situs layanan *web* yang dikomersilkan yang memungkinkan perusahaan kecil dan individu untuk menyewa komputer atau *server*, agar dapat menjalankan aplikasi komputer mereka. Batu lompatan terbesar berikutnya terjadi pada tahun 2009 dengan *Web 2.0* mencapai puncaknya. *Google* dan lainnya memulai untuk menawarkan aplikasi *browser-based* untuk perusahaan besar seperti *Google Apps*. *Cloud computing* sudah menyebar luas saat ini hingga kepada pengguna *Google Doc*. *Smartphone* ikut mengikuti perkembangan ini salah satunya android. *Smartphone* berbasis android ini semakin berkembang tidak hanya untuk layanan telepon ataupun mengirim pesan tapi sudah mencakup akses internet maupun sudah mendukung aplikasi-aplikasi yang dapat membantu kegiatan manusia dalam kehidupan sehari-hari.

Aplikasi *Poin of sale (PoS)* adalah aplikasi yang digunakan untuk membantu proses transaksi dalam bidang penjualan. Aplikasi ini banyak dikembangkan pada desktop atau komputer, ada juga sebagian yang sudah menggunakan *mobile smartphone*. Namun aplikasi-aplikasi tersebut masih bersifat *local database* atau media penyimpanan masih didalam *hardisk* komputer atau didalam memori *smartphone* tersebut. Hal ini sangat berbahaya jika suatu saat data-data transaksi tersebut hilang akibat kerusakan *hardware*. Sedangkan pengguna atau pemilik usaha ingin melihat atau mengambil data yang terdahulu guna kepentingan laporan.

Dari keadaan diatas jika mengkorelasikan dengan perkembangan teknologi di bidang *Smartphone* dan *cloud computing* maka tercipta gagasan baru. *Smartphone* yang semakin canggih dimungkinkan dapat mendukung kegiatan PoS tersebut. Sedangkan media penyimpanan data dapat di *back up* dengan teknologi *cloud (PaaS)* sehingga dapat

mengurangi kehilangan data yang disebabkan kerusakan *hardware*. Selain itu pemilik usaha juga dapat melihat perkembangan usahanya dimana pun berada dengan syarat harus terkoneksi dengan internet.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan sebagai berikut :

1. Bagaimana merancang Aplikasi PoS yang dapat berjalan di Android.
2. Bagaimana Aplikasi dapat menyimpan data di *Cloud*.

1.3 Batasan Masalah

Dari rumusan masalah yang ada mempunyai batasan masalah seperti berikut:

1. Hanya berjalan di *platform* Android.
2. Aplikasi hanya bisa berjalan jika terkoneksi dengan internet.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah :

1. Dapat mengimplementasikan Aplikasi PoS yang dapat berjalan di *platform* Android
2. Dapat mengimplementasikan media penyimpanan di *cloud*

1.5 Sistematika Penulisan

BAB I : PENDAHULUAN

Bab ini di uraikan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA

Bab ini di uraikan tentang pengertian Android, basis data, MySQL, Cloud, bahasa pemrograman java, bahasa pemrograman XML, Elipse.

BAB III : DISKRIPSI PERANGKAT LUNAK

Bab ini di uraikan tentang identitas perangkat lunak, fungsi perangkat lunak dan cara kerja perangkat lunak

BAB IV: PEMBAHASAN

Bab ini di uraikan tentang cara Android PoS Cloud-based tersebut berjalan.

BAB V: KESIMPULAN

Bab ini memuat kesimpulan dari pembahasan dalam BAB III dan saran.

BAB 2

LANDASAN TEORI

2.1 Pengertian PoS

Menurut Novita dan Edin Surdi dalam skripsi yang berjudul “Perancangan sistem informasi PoS pada PD Tokyo”,2013, Point of Sale atau biasa disingkat PoS merupakan kegiatan yang berorientasi pada penjualan serta sistem yang membantu proses transaksi. Setiap PoS terdiri dari hardware dan software dimana kedua komponen tersebut digunakan untuk setiap proses transaksi. PoS software merupakan komponen utama dari sistem PoS yang pada akhirnya menentukan jalannya proses, seperti apa yang harus dilakukan dan bagaimana harus melakukan. Sedangkan hardware PoS dibutuhkan untuk menjalankan fungsinya, membantu proses pembayaran. Aplikasi ini biasanya digunakan untuk minimarket atau dirumah makan dengan diinstallkan pada komputer.

2.2 Pengertian Android

Menurut Nazruddin Safaat H, dalam buku yang berjudul “Android” (2012), Android adalah sebuah sistem operasi yang berbasis *linux* sebagai kernelnya. Android digunakan pada telepon seluler seperti telepon pintar(*smart phone*) dan komputer *tablet*. Android begitu pesat berkembang di era saat ini, salah satu faktor yang mempengaruhi yaitu Android menyediakan *platform* terbuka (*Open Source*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Android menjadi sangat terkenal setelah dibeli oleh perusahaan *search engine* terbesar, yaitu Google Inc. Android didirikan oleh *Andy Rubin, Rich Milner, Nick Sears* dan *Cris White* pada tahun 2003. Pada saat perilisannya pertama android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standard terbuka pada perangkat seluler.

Android memiliki dua distributor, yaitu *Google Mail Service (GMS)* dan *Open Handset Distributor(OHD)*. *GMS* adalah distributor Android yang mendapat dukungan penuh dari Google, sedangkan *OHD* adalah distributor Android tanpa dukungan langsung dari Google. Pesatnya pertumbuhan Android karena Android adalah *platform* yang sangatlah lengkap, baik dari segi sistem operasi, aplikasi, *tools* pengembangan, dan *market* yang menyimpan berbagai aplikasi serta ditambah dengan berbagai dukungan oleh komunitas *Open Source* di dunia.

2.2.1 Android SDK (Software Development Kit)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman *Java*. Untuk *source* SDK Android dapat dilihat dan di unduh langsung disitus resmi pengembang SDK android di <http://www.developer.android.com>

2.2.2 ADT (Android Development Tools)

Android Development Tools (ADT) adalah *plugin* yang di *desain* untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Eclipse. ADT akan memudahkan *developer* untuk membuat proyek, GUI (*Graphical User Interface*) Aplikasi, dan menambahkan komponen yang lain. Untuk mengunduh ADT dapat dilakukan di <http://developer.android.com/sdk/eclipse-adt.html>.

2.3 Pengertian Cloud

Menurut Ryan A.P.Armanda, dalam skripsinya (2010), *Cloud Computing* atau komputasi awan adalah gabungan pemanfaatan teknologi komputer dan pengembangan berbasis *Internet*. Suatu metode komputasi di mana kapabilitas terkait teknologi informasi disajikan sebagai suatu layanan sehingga pengguna dapat mengakses lewat *Internet* tanpa mengetahui apa yang ada didalamnya. Dengan teknologi ini pengguna komputer ataupun *Smart Phone* dapat menggunakan *source* tanpa harus membeli dan memiliki di dalam komputer maupun *Smart Phone*. *Cloud Computing* mulai dikenal sejak tahun 1960, dimana pada masa itu terjadi revolusi komputer, IBM salah satunya yang merupakan pengembang terbesar dalam bidang teknologi ini.

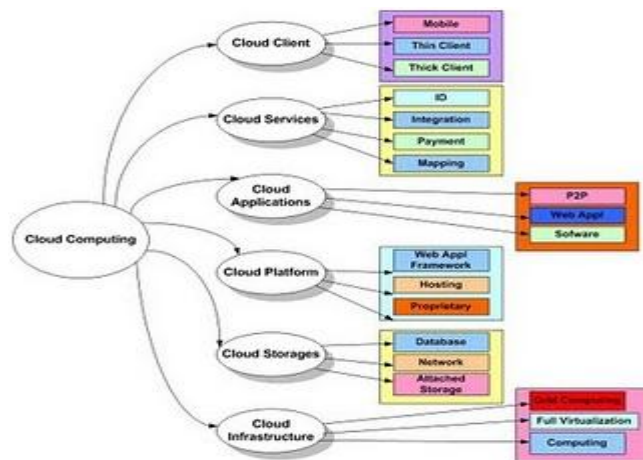
Pada dekade tahun 1970, *Tymeshare* sebagai penyedia jasa layanan mengembangkan sistem penyewaan ruang *mainframe* yang dapat diakses melalui jalur telepon. Perusahaan tersebut menyediakan layanan komputer tingkat tinggi kepada perusahaan besar saat itu. Pada tahun 1980 dan 1990 lahirlah *Application Services Provider*(ASP), yang pada masa itu dikenal sebagai layanan pihak ketiga yang berfungsi sebagai pedelegasian data, manajemen data, dan hosting *software* dalam layanan berbasis sewa. ASP melahirkan kesempatan untuk perusahaan berskala kecil untuk menggunakan teknologi terbaru, dengan biaya yang relatif rendah.

Cloud Computing memiliki dua konsep, yakni konsep pada aplikasi yang dijalankan sebagai *service* dengan bantuan *Internet*, dan konsep perangkat keras yang berada pada

pusat *data center* yang menyediakan akses bagi konsep sebelumnya. Konsep aplikasi sebagai sebuah *service*, telah lama dikenal dengan sebutan *Software as a Service (SaaS)*. *Hardware* yang berada pada *data center* dan menjalankan aplikasi tersebut disebut dengan *Cloud*.

2.4 Arsitektur Cloud Computing

Arsitektur yang unik dari *cloud computing* memiliki peranan yang cukup besar pada proses tercipta dan berkembangnya teknologi ini. Penerapan *Software as a Service* dan *client as a service* memiliki dampak yang sangat besar kepada tingginya harapan terhadap perkembangan teknologi ini.



Gambar 1. Struktur Cloud Computing

Dari gambar diatas dapat dilihat, ada beberapa arsitektur dasar dari sistem *cloud computing* secara umum. Diantaranya adalah *cloud client*, *cloud services*, *cloud Applications*, *cloud platform*, *cloud storage*, dan *cloud infrastrukture*

2.5 Mekanisme Cloud Computing

2.5.1 Software as a Service (SaaS)

Layanan yang menyediakan aplikasi siap jadi kepada *end user*. Ciri dari layanan ini adalah *user* tidak perlu membuat aplikasi dan *infrastruktur*. Contoh dari layanan ini adalah *gmail*, *ymail*, *facebook*, *twitter*, *Dropbox*. Konsep ini dapat didefinisikan sebagai proses *deliver* dari sebuah aplikasi melalui internet. Konsep SaaS mengeliminasi kebutuhan untuk menginstall dan menjalankan sebuah aplikasi dari sisi pengguna. Konsep ini juga mengurangi beban pengguna dalam hal aplikasi dan *support*. Sisi negatifnya terletak pada hilangnya kemampuan pengguna untuk merubah versi *software*. Aplikasi seperti CRM,

Video Convergence, sumber daya manusia, *IT service management*, *accounting*, keamanan IT, dll. Merupakan pilar kesuksesan SaaS

2.5.2 Platform as a Service (PaaS)

Platform as a Service (PaaS) adalah sebuah layanan komputasi dimana seluruh fasilitas yang dibutuhkan untuk memenuhi seluruh *life cycle* dari sebuah aplikasi, yaitu membangun aplikasi, melakukan tes aplikasi, dan finalisasi serta penyelesaian sebuah aplikasi dalam jaringan internet dari sebuah *cloud* tanpa membutuhkan proses *download* dan instalasi *software* dari pihak pengembang dan pengguna. Layanan ini menyediakan tempat untuk menjalankan aplikasi dari *user*. Tempat yang dimaksud mencakup Sistem Operasi, *Database*, *Framework*, dsb yang merupakan wadah untuk berjalannya aplikasi. Ciri dari layanan ini adalah *user* tidak perlu melakukan *Maintenance* dan tidak perlu menyiapkan *infrastruktur*. Sehingga *user* dapat tetap fokus membangun aplikasinya. Contoh dari layanan ini adalah *Windows Azure*, *GoogleApps Engine*.

2.5.3 Infrastructure as a Service (IaaS)

Layanan yang menyewakan *infrastruktur* IT kepada *user* yang ingin membangun layanan *cloud*. *Infrastruktur* disini bersifat fisik, bisa berupa *memori*, *server*, jaringan dsb. Sedangkan pembuatan aplikasi dan *konfigurasi* di serahkan kepada *user*. *Cloud provider* hanya menyediakan infrastruktur berdasarkan *request* dari *user*. Ciri dari layanan ini adalah jika *user* ingin *upgrade* memory atau menambah *server*, *user* tinggal menghubungi *provider*, kemudian *provider* akan menyediakan sesuai permintaan. Contoh dari layanan ini adalah *Amazon EC2*, *Rackspace cloud*. IaaS dapat berjalan pada klien dengan bantuan APIs. Pada umumnya *RESTful* dan *SOAP APIs* *software* digunakan untuk mengakses infrastruktur ini. IaaS sangat bergantung pada teknologi *virtualization* pada platform untuk menjalankan sebuah *virtual machine* tertentu.

Berbagai penyedia layanan ini membedakan layanan mereka pada jumlah Sistem Operasi pada *platform* yang mereka *support*, *software* yang diinstalasi pada Sistem Operasi, harga dan *service agreement*.

2.6 Pengertian Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform*. Eclipse pada

saat ini merupakan salah satu IDE yang disenangi karena gratis dan *open source*, yang berarti setiap orang dapat melihat kode pemrograman perangkat lunak ini. Berikut sifat dari eclipse :

- A. *Multi Platform* → *Windows, Linux, Mac OS X, Solaris.*
- B. *Multi Language* → *JAVA, C/C++, Cobol, PHP, Python, Perl.*
- C. *Multi Role* → Pengembangan perangkat lunak, Tes perangkat lunak, Pengembangan Web, dan Dokumentasi.

2.7 Pengertian Java

Menurut Abdul Kadir dalam bukunya yang berjudul “ Dasar Pemrograman Java 2”(2004), *Java* adalah bahasa pemrograman serbaguna. *Java* dapat digunakan untuk membuat suatu program sebagaimana membuat program di **Pascal** maupun **C language**. Yang lebih menarik, *Java* juga mendukung sumber daya Internet yang saat ini populer, yaitu *Word Wide Web* atau yang sering disebut *Web* saja. *Java* juga mendukung aplikasi *klien/server*, baik dalam jaringan lokal (LAN) maupun jaringan berskala luas (WAN).

Java dikembangkan oleh Sun Microsystems pada Agustus 1991, dengan nama semula *Oak*. Konon *Oak* adalah pohon semacam Jati yang terlihat dari jendela tempat pembuatannya, **James Gosling**, bekerja. Ada yang mengatakan *Oak* adalah singkatan dari “*Object Application Kernel* “, tetapi ada yang menyatakan hal itu muncul setelah nama *Oak* diberikan. Pada Januari 1995, Karena nama *Oak* dianggap kurang komersil, maka diganti menjadi *Java*.

Dalam sejumlah literatur *Java* disebutkan bahwaI merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman, yaitu *C*, *C++*, *Object-C*, *SmallTalk*, dan *Common LISP*. Selain itu *Java* juga dilengkapi dengan unsur keamanan. Yang tak kalah penting adalah bahwa *Java* menambah paradigma pemrograman yang sederhana. Selain itu *Java* juga tidak bergantung pada Platform, yang berarti *Java* dapat dijalankan pada sembarang komputer bahkan pada sembarang Sistem Operasi.

2.8 Pengertian UML

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat

lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP).

Bahasa UML digunakan untuk menentukan, *memvisualisasikan*, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek, namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Hal tersebut membuat penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

UML menyediakan beberapa jenis diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

- a. *Use Case Diagram* untuk memodelkan proses bisnis. *Use case diagram* digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk *elips* dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.
- b. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
- c. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*. *Sequence diagram* menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.
- d. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*. *Collaboration diagram* dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence diagram* yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration diagram* lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh *object*.

- e. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
- f. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
- g. *Class Diagram* untuk memodelkan struktur kelas. II-23
- h. *Object Diagram* untuk memodelkan struktur *object*.
- i. *Component Diagram* untuk memodelkan komponen *object*.
- j. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

2.9 Pengertian PHP

Menurut Stendy, B Syakur, PHP adalah singkatan dari “*Personal Home Page*” atau biasa disebut juga “*Hypertext Preprocessor*”, merupakan sebuah bahasa *scrip* tingkat tinggi yang dipasang pada dokumen HTML (*Hiper Text Markup Language*). *Syntax* dalam PHP hampir sama dengan bahasa C, *java*, dan *perl*. Tujuan penggunaan PHP adalah untuk merancang web yang dinamis dan dapat bekerja secara otomatis.

PHP pertama kali dibuat oleh *Ramus Lerdroft*, seorang *programer C*. Pada waktu itu PHP bernama FI (*Form Interpreter*) yang wujudnya sekumpulan *scrip* yang digunakan untuk mengolah data *form* dari web.

PHP dirilis menjadi *open source* pada tahun 1995. Setelah tahun tersebut banyak *programer* yang tertarik untuk ikut mengembangkan. Pada tahun 1996 PHP versi 2.0 telah dirilis yang mempunyai kemampuan untuk mengakses *database* dan dapat terintegrasi dengan HTML. PHP versi 4.0 keluar pada tanggal 22 Mei 2000, pada versi ini ditambahkan *Zend Engine* yang dibuat oleh *Zend Suaski* yang merupakan penyempurnaan dari PHP *scrip engine*. Pada Juni 2004 PHP mengeluarkan versi terbaru yaitu PHP5.0. versi ini memasukan model Program Berorientasi *Objek* ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi *objek*.

2.10 Pengertian MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data atau DBMS yang *multitread*, *multi-user*. MySQL perangkat lunak gratis dibawah lisensi *GPL (General Public Licence)*.

MySQL awalnya diciptakan pada tahun 1979, oleh “*Michael Monty Widenius*”, seorang *programer* komputer asal Swedia. *Monty* mengembangkan sistem *database* sederhana yang dinamakan *UNIREG* yang menggunakan koneksi *low-level ISAM database engine* dengan *indexing*. TcX tempat kerja *Montly* mulai mengembangkan *UNIREG* sebagai sistem *database*, namun *UNIREG* dianggap tidak cocok. Akhirnya TcX menggunakan *mSQL(miniSQL)*.

Pada May 1995, sebuah RMBMS baru yang dinamakan MySQL dirilis. MySQL adalah penggabungan antara *UNIREG* dengan *mSQL* yang dikembangkan oleh *Montly* dan *David Hughes*. *Source Code* pada MySQL dapat digunakan secara bebas.

MySQL dapat diakses dengan berbagai bahasa pemrograman diantaranya : *bahasa C, C++, Java, Perl, PHP, Python* dll. Semua bahasa pemrograman mempunyai fasilitas untuk berkomunikasi dengan basis data MySQL menggunakan ODBC.

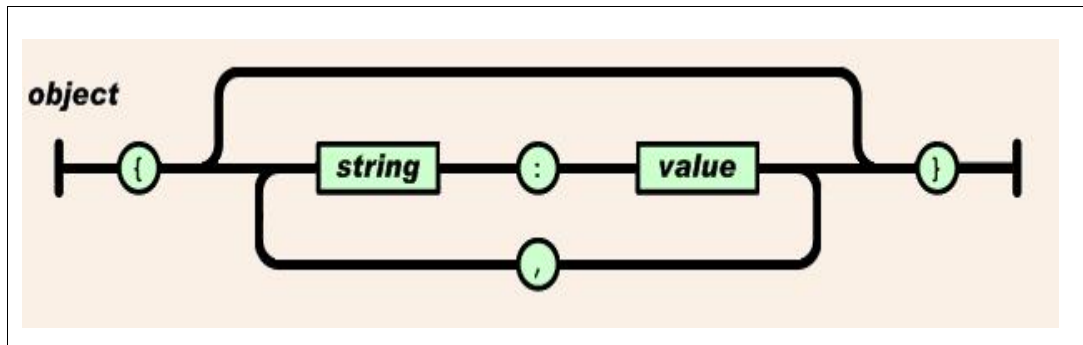
2.11 Pengertian JSON

Diambil dari situs resmi JSON, JSON(*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman *Java Script*. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, Java Script, Perl, Python PHP, dll. Karena sifat tersebut menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terbat dari dua struktur yaitu :

- A. Kumpulan pasangan nama/ nilai.pada beberapa bahasa,hal ini dinyatakan sebagai objek, rekaman, struktur, kamus, table hash, daftar kunci, atau array
- B. Daftar nilai terurutkan. Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik, vector, daftar, atau uruta.

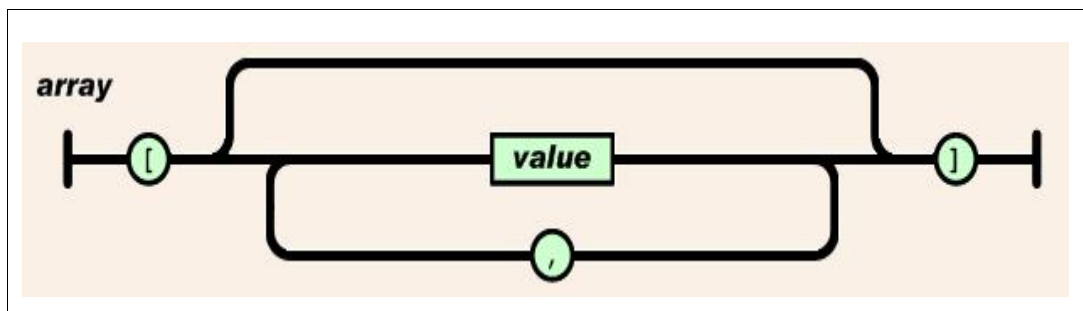
Stuktur – struktur data ini disebut sebagai struktur data universal. Pada dasarnya,semua bahasa pemrograman modern mendukung struktur data dalam bentuk yang sama atau dalam bentuk yang berlainan. JSON sendiri mengguakan bentuk struktur data seperti berikut ini.

- A. Objek adalah sepasang nama/nilai yang tidak terurut, Objek dimulai dengan {(kurung kurawal buka) dan diakhiri }(kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan seriap pasang nama/nilai dipisahkan oleh ,(koma).



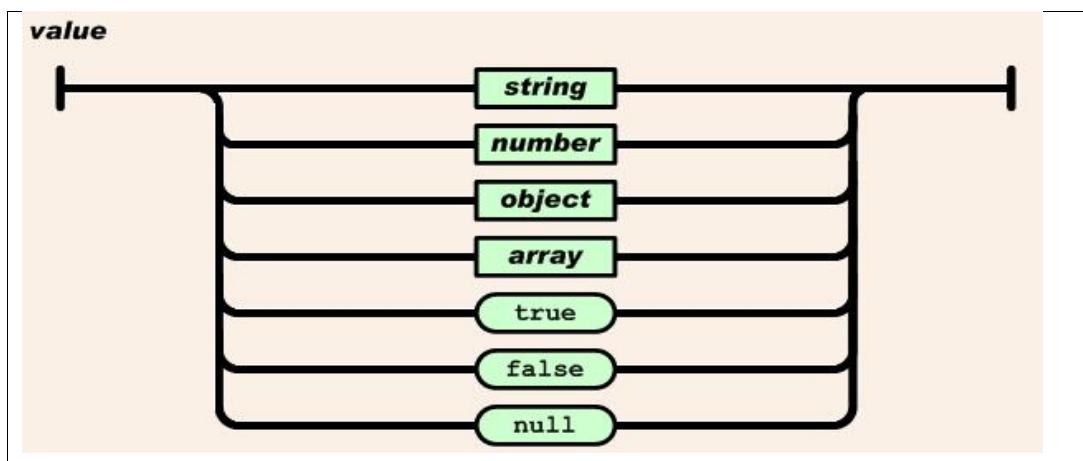
Gambar 2. Objek dalam JSON

- B. Larik adalah kumpulan nilai yang terurutkan. larik dimulai dengan dan diakhiri dengan tanda “[]”, dan setiap nilai dipisahkan dengan koma



Gambar 3. Array dalam JSON

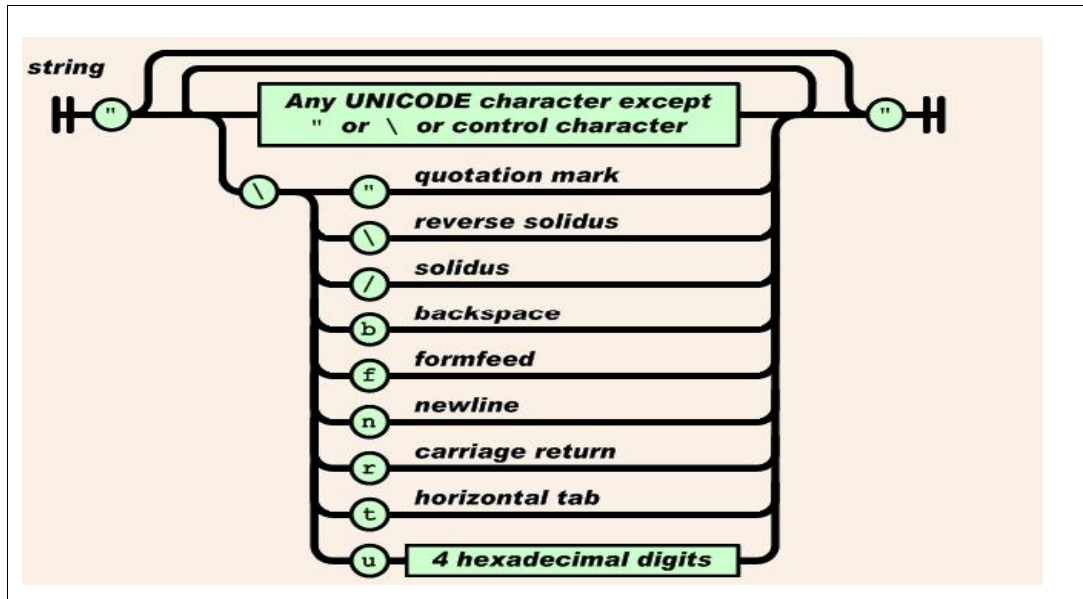
- C. Nilai dapat berupa sebuah string dalam tanda kutip ganda ,angka, *true*, *false*,objek, *array*. Struktur tersebut dapat dibuat bertingkat seperti gambar berikut.



Gambar 4. Nilai dalam JSON

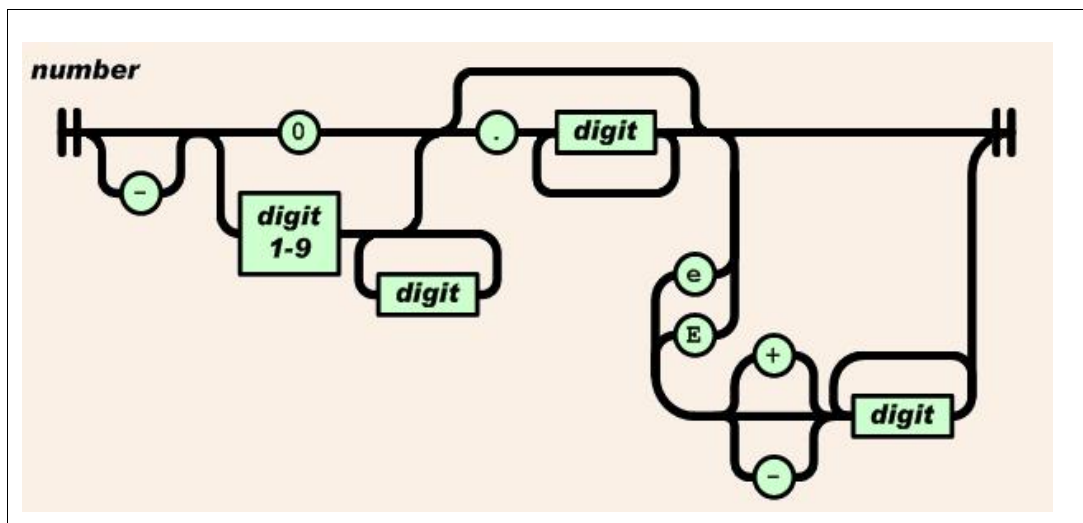
- D. String adalah kumpulan dari nol atau lebih karakter Unicode,yang dibungkus dengan tanda kutip ganda. Didalam string dapat digunakan *backsash escapes* “\”

untuk membentuk karakter khusus. Sebuah karakter yang mewakili karakter tunggal pada string.berikut gambaran string untuk JSON



Gambar 5. String dalam JSON

E. Angka adalah sangat mirip dengan angka di pemrograman C atau Java, kecuali format octal dan heksadesimal. Berikut format angka dalam JSON



Gambar 6. Angka dalam JSON

BAB 3

ANALISA DAN PERANCANGAN

3.1 Analisa

Pada bab ini akan menjelaskan tentang analisa dan perancangan Android PoS Cloud-based yang dikembangkan pada *platform* Android dan media penyimpanan data berupa *cloud*

3.1.1 Analisa Masalah

Studi kasus yang dilakukan adalah kantin Politeknik Negeri Batam. Permasalahan yang ada saat ini adalah pelayanan kantin yang masih manual yaitu, pembeli datang ke vendor untuk memesan, kemudian vendor membuat *struk* untuk pembayaran dikasir, pembeli akan membayar ke kasir dengan memberikan *struk* ke vendor. Setelah membayar pembeli akan kembali ke vendor untuk menyerahkan kwitansi pembayaran guna mendapat pesanan yang dibeli. Keadaan tersebut memungkinkan terjadi kesalahan dan tidak *efisien*, karena pembeli harus bolak-balik ke vendor untuk mengambil pesanan. Yang kedua media penyimpanan data masih bersifat manual data ditulis pada kertas biasa. Keadaan tersebut sangat rentan kehilangan data. Sementara data-data tersebut akan digunakan dikemudian hari.

3.1.2 Analisa Kinerja Sistem Secara Umum

Aplikasi ini dapat digunakan oleh *user* melalui internet. *User* yang menggunakan aplikasi ini dapat dibedakan menjadi dua yaitu, Kasir, dan Vendor. Kasir mendapat hak akses, *create*, *update*, *delete user*, menu, laporan serta mendapat akses melakukan pembayaran. Cara kerjanya, pembeli akan datang ke vendor untuk memesan, vendor akan menginputkan pesanan, nama vendor, nomor pesanan, jumlah pesanan ke *database*, setelah diinput secara otomatis sistem akan memberikan nomor pesanan guna pembayaran. Pembeli membayar ke kantin dengan memberi informasi nomor pesanan. Kasir akan melihat pesanan dan memberitak nominal yang harus dibayar oleh pembeli, jika sudah dibayar kasir akan menyimpan data pembayaran kedalam cloud berupa jumlah pembayaran, hari, dan nama vendor.

3.2 Analisa Kebutuhan Sistem

Kebutuhan non fungsional yang dibutuhkan untuk membangun sistem ini terdiri dari dua hal, yaitu :kebutuhan *hardware* dan kebutuhan *software*.

3.2.1 Analisa Kebutuhan *Hardware*

Untuk merancang aplikasi dibutuhkan *hardware* yang mendukung aplikasi tersebut.

Hardware yang dibutuhkan yaitu :

Table 1. Hardware Komputer

Hardware	Speck
<i>Processor</i>	Intel(R) Core(TM) i5 2,75GHz
<i>Memory</i>	4,00 Gb DDR3
<i>Hard Drive</i>	320 Gb SATA
<i>Video Card</i>	ATI Radeon HD 4300
<i>Display</i>	14.0" HD LED LCD
<i>Audio</i>	Realtek High Defenition Audio

3.2.2 Analisa Kebutuhan *Software*

Software yang akan digunakan untuk membuat aplikasi Android PoS Cloud-based meliputi :

Table 2. Kebutuha Software Pengembangan

Software	Speck
<i>Sistem Operasi</i>	Windows 7 32 Bit
<i>Building Tool</i>	ADT (Android Development Tools)
	SDK (Software Development Kit)
	IDE Ecips
<i>Implementasi</i>	AVD(Android Virtual Device) Versi 3.0

3.2.3 Analisa Kebutuhan SDM(Sumber Daya Manusia)

Sumber daya manusia yang dibutuhkan untuk menggunakan aplikasi ini meliputi :

- a. Kasir
- b. Vendor

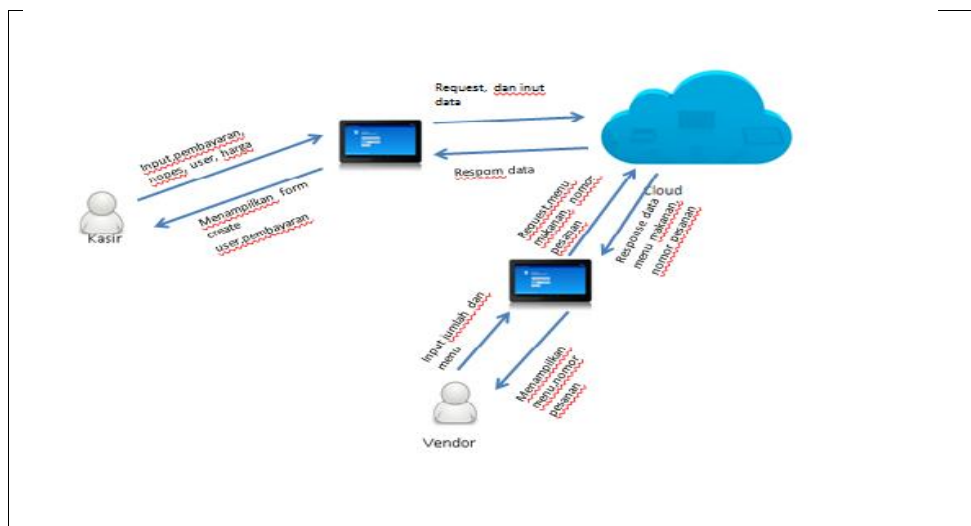
3.2.4 Analisa Kebutuhan Fungsional

Kebutuhan fungsional pada aplikasi Android PoS Cloud-based adalah :

- a. Vendor dan Kasir dapat *login* ke sistem
- b. Aplikasi dapat melihat, menambah, mengedit, dan menghapus data *user*
- c. Aplikasi dapat melihat, menambah, mengedit, dan menghapus data harga.
- d. Aplikasi dapat melakukan proses pemesanan
- e. Aplikasi dapat memproses pembayaran.
- f. Aplikasi dapat menampilkan laporan.

3.4 Diskripsi Umum Sistem

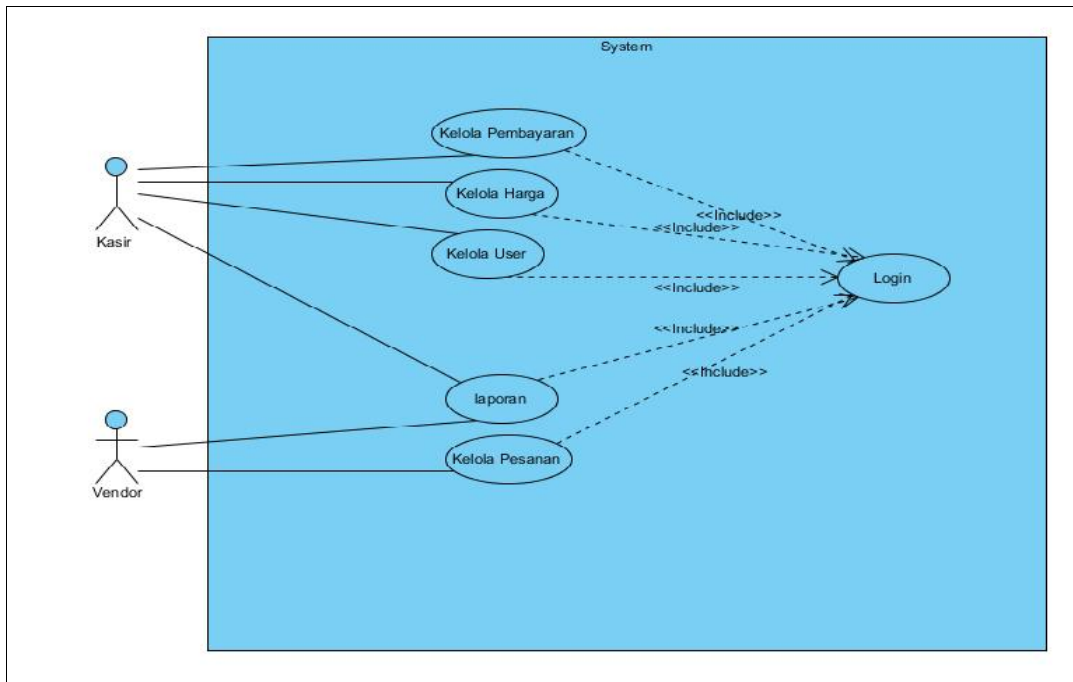
Dalam Tugas Akhir ini, penulis mencoba membangun aplikasi *Android PoS Cloud-based* yang mampu memberikan penanganan terhadap kegiatan pembayaran dengan memanfaatkan teknologi *Smart Phone* khususnya Android dan teknologi *Cloud*. Cara kerjanya adalah pertama kasir akan melakukan CRUD(*create, Read, update, dan delete*) untuk *user, menu, dan harga* kemudian disimpan dalam *cloud*. *User* kasir juga akan melakukan transaksi pembayaran. *User* vendor mempunyai tampilan berupa menu masakan, dan laporan. Proses kerjanya adalah vendor/pembeli akan memilih menu yang akan dipesan, setelah dipilih vendor diminta untuk memasukkan jumlah pesanan. Selanjutnya vendor diminta untuk menekan *bottom* pesan maka system akan memberikan nomor pesanan guna untuk pembayaran. Setelah nomor pesanan didapat, pembeli diharuskan datang kekasir untuk melakukan pembayaran dengan memberikan informasi nomor pesanan yang didapat. Dengan nomor pesanan yang didapat kasir akan melakukan pengecekan total pembayaran yang harus dibayar oleh pembeli. Setelah semua proses dilakukan vendor maupun kasir dapat melihat laporan pendapatan guna menyamakan hasil yang didapat dengan cara memilih menu laporan, memasukkan nama vendor, tanggal awal dan tanggal akhir. Setelah semua diisi langkah selanjutnya *user* diminta untuk menekan tombol cek, secara otomatis system akan memberika informasi total pendapatan beserta nama vendor.



Gambar 7. Diskripsi Umum Sistem

3.5 Diagram Use Case Android PoS Cloud-based

Diagram use case ini akan menggambarkan cara kerja Android PoS Cloud-based.



Gambar 8. Use Case Android PoS Cloud

3.6 Definisi Aktor

Berikut ini adalah deskripsi pendefinisian aktor pada Aplikasi Android PoS Cloud-based

Table 3. Diskripsi Aktor

No	Aktor	Deskripsi
1	Kasir	Aktor akan melakukan kegiatan pembayaran, kasir akan menanyakan nomor pesanan kemudian akan melihat pesanan beserta harga selanjutnya akan menyimpan transaksi pembayaran kedalam <i>cloud</i> , disisi lain <i>user</i> kasir juga mendapat akses untuk CRUD <i>user</i> , menu, harga
2	Vendor	Aktor akan menerima pesanan selanjutnya akan menyimpan pesanan id vendor, jumlah pesanan, menu dan jumlah pesanan

3.7 Definisi Use Case

Berikut diskripsi dari definisi *use case* pada Aplikasi Android PoS Cloud-based.

Table 4. Definisi Use Case Android PoS Cloud-based

No	Use Case	Diskripsi
1	<i>Login</i>	Merupakan proses pengecekan hak akses siapa yang berhak memasukan (<i>input</i>) <i>user</i> dan menu dan harga, <i>login</i> wajib untuk fungsi- fungsi yang berkaitan dengan akses penambahan ke dalam Basis Data oleh sebab itu <i>login</i> ini di perlukan untuk mengecek validasi <i>User</i> .
2	Kelola Pembayaran	Merupaka proses <i>insert</i> data kedalam database setelah transaksi di proses. Transaksi yang dimaksud adalah ketika pembeli melakukan pembayaran terhadap item yang telah dibeli dengan melakukan pengecekan sebelumnya kedatabase
4	Kelola Pesanan	Memasukan pesanan yang dipilih kedalam database guna proses pembayaran.
5	Kelola <i>User</i>	Kasir akan membuat, edit, maupun delete <i>user</i> kasir dan Vendor
6	Kelola Harga	kasir akan membuat, edit, maupun delete harga menu, jenis dan id
7	Laporan	<i>User</i> kasir dan <i>user</i> vendor akan mendapat tampilan laporan berupa berapa total pesanan yang didapat dengan parameter tanggal dan nama <i>user</i>

3.8 Skenario Use Case

Skenario *Use Case* adalah penjelasan secara detail dari *use case* diagram. Berikut skenario *use case* untuk Android PoS Cloud-based.

3.8.1 Nama Use Case : *login*

Nama *Use Case* : *login*

Actors : Kasir, Vendor

Deskripsi :

User memasukan *username* dan *password*.Sistem melakukan verifikasi ke dalam *database*.

Kondisi Awal : *user* membuka aplikasi Andrid PoS cloud-based

Kondisi Akhir : Home dari masing-masing *user*.

Table 5. Skenario Use Case *login*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukan <i>username</i> dan <i>password</i>	
	2. <i>Validasi</i> ke dalam <i>database</i> dari data yang dimasukkan
	3. Masuk ke Home <i>user</i>

3.8.2 Nama Use Case :Kelola Pembayaran

Nama Use Case : Pembayaran

Actors : Kasir

Deskripsi :

User melakukan aktifitas pembayaran setelah pembeli memesan makan, kemudian user akan menyimpan data pembayaran ke dalam cloud

Kondisi awal : User sudah login dan sudah masuk menu Pembayaran.

Kondisi akhir : User menyimpan data transaksi kedalam cloud

Table 6. Use Case Pembayaran

Aksi Aktor	Reaksi Sistem
Skenario Normal	
2. Memasukan nomor pesanan	
	4. Menampilkan menu yang dipesan, nama vendor, jumlah, harga permenu serta jumlah yang harus dibayar
5. Menekan botton bayar	
	8. Menyimpan data dan memberikan notifikasi berhasil disimpan

3.8.3 Nama Use Case :Kelola Pesanan

Nama Use Case : Pesanan

Actors : Vendor

Deskripsi :

User akan menyimpan menu yang dipilih oleh pembeli ke dalam database.

Kondisi Awal : User sudah login dan sudah masuk menu pesanan

Kondisi Akhir : pembayaran berhasil disimpan kedalan database.

Table 7. Use Case Pesanan

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. user memasukan jumlah pembelian	
2. Memyimpan menu, nama vendor,harga per menu,dan jenis menu.	
	3. Menyimpan kedalam database
	4. Kembali ke list menu

3.8.4 Nama Use Case :Kelola User

Nama Use Case :Kelola User

Actors :Kasir

Deskripsi :

User Kasir akan *login*, setelah *login* akan memasukan inputan berupa detail *user* Vendor mauun kasir.

Kondisi awal : *User* sudah *login* dan sudah masuk menu kelola *user*

Kondisi akhir : *user* baru sudah di buat.

Table 8. Use Case Kelola User

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu tambah <i>user</i>	
2. Memasukan nama <i>user</i> , bagian, <i>password</i> , dan hp	
3. Menyimpan data ke <i>database</i>	
	4. Menyimpan kedalam <i>databae</i>
	5. kembali ke menu buat <i>user</i>
6. Memilih menu <i>edit user</i>	
	7. Menampilkan menu <i>edit</i>
8. Memasukan id <i>user</i>	
	9. Menampilkan Informasi <i>User</i>
10. Update Informasi <i>User</i>	
11. Menyimpan kedala <i>database</i>	
	12. Menyimpan data kedalam <i>database</i>
	13. Kembali ke menu buat <i>user</i>
14. Memilih menu delete <i>user</i>	
	15. Menampilkan menu <i>delete</i>
16. Memasukan id <i>user</i>	
17. Delete <i>user</i>	
	18. update <i>database</i>
	19. kembali ke menu buat harga

3.8.5 Nama Use Case :Kelola Harga

Nama Use Case : Kelola Harga

Actors : Kasir

Deskripsi :

User Kasir akan *login*, setelah *login* akan memasukan inputan harga, menu, dan id

Kondisi awal : *User* sudah *login* dan sudah masuk menu kelola Harga.

Kondisi Akhir : Harga dan menu baru sudah dibuat.

Table 9. Use Case Kelola Harga

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu buat harga	
	2. Menampilkan menu buat harga
3. memasukkan harga dan menu	
4. Menyimpan kedalam <i>database</i>	
	5. Menyimpan kedalam <i>database</i>
	6. kembali ke menu buat harga
7. Pilih <i>edit</i> harga	
	8. Menampilkan menu <i>edit</i> harga
9. Memasukkan nama menu	
	10. Menampilkan menu yang dicari
11. Edit informasi menu	
12. Menyimpan data menu	
	13. menyimpan data menu kedalam <i>database</i>
	14. kembali ke menu buat harga
15. Pilih <i>delete</i> harga	
	16. menampilkan menu <i>delete</i> harga
17. Memasukan nama menu yang akan di <i>delete</i>	
	18. Menampilkan menu yang dicari.
19. <i>Delete</i> menu	
	20. <i>update database</i>
	21. Kembali menu buat harga

3.8.5 Nama Use Case :Laporan

Nama Use Case : Laporan

Actors : Kasir, vendor

Deskripsi :

User akan memasukkan id vendor dan tanggal , setelah memasukan user akan mendapat tampilan hasil penjualan.

Kondisi awal : User sudah *login* dan masuk ke dalam menu utama

Kondisi Akhir : user melihat laporan berdasarkan hari dan id vendor

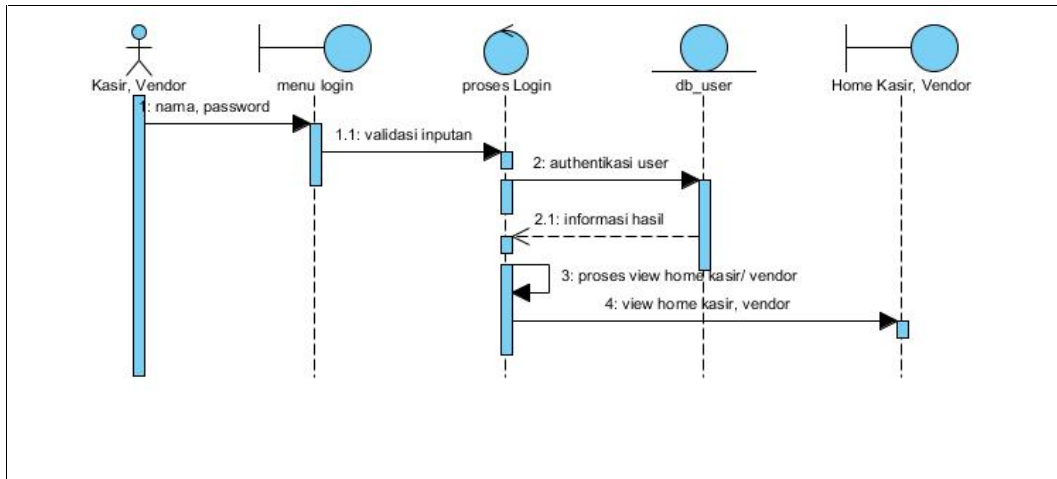
Table 10. Use Case Laporan

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>user</i> memasukkan tanggal dan id vendor	
2. <i>User</i> akan mengklik <i>button</i> cari	
	3. sistem mencari kedalam <i>database</i> berdasarkan <i>parameter</i>
	4. Menampilkan Laporan

3.9 Sequence Diagram Android PoS Cloud-based

Sequence Diagram digunakan untuk menggambarkan skenario atau rancangan langkah-langkah yang dilakukan sebagai respon dari sebuah *even* untuk menghasilkan *output* tertentu.

3.9.1 Sequence Diagram login

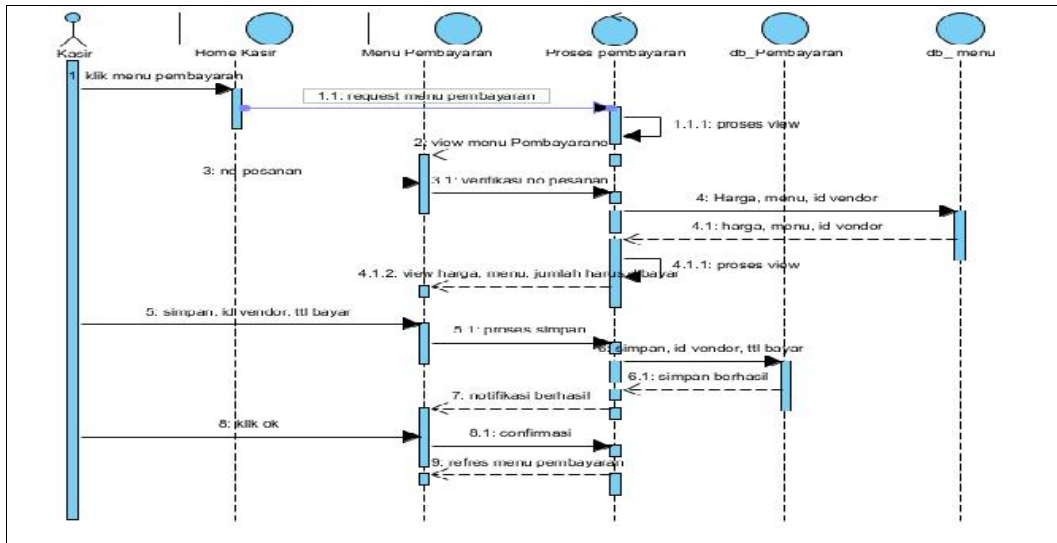


Gambar 9. Sequence Diagram login

a. Penjelasan Sequence Diagram login

User memasukkan *username* dan *password* kemudian sistem akan validasi di *database*, jika sesuai maka akan dilanjutkan ke halaman utama sesuai dengan *akses* yang didapat

3.9.2 Sequence Diagram Kelola Pembayaran

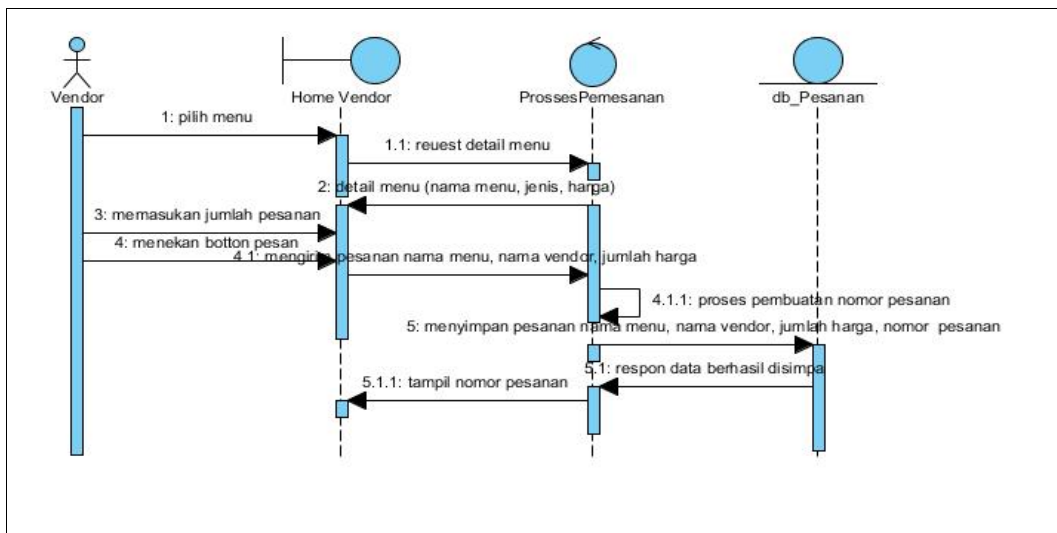


Gambar 10. Sequence Diagram Kelola Pembayaran

a. Penjelasan Sequence Diagram Kelola Pesanan

User memasukkan nomor pesanan kemudian sistem akan menampilkan menu yang dibeli, total yang harus dibayar, nomor pesanan, dan nama vendor,. Setelah dibayar user akan menyimpan transaksi ke *batabase*

3.9.3 Sequence Diagram Kelola Pesanan



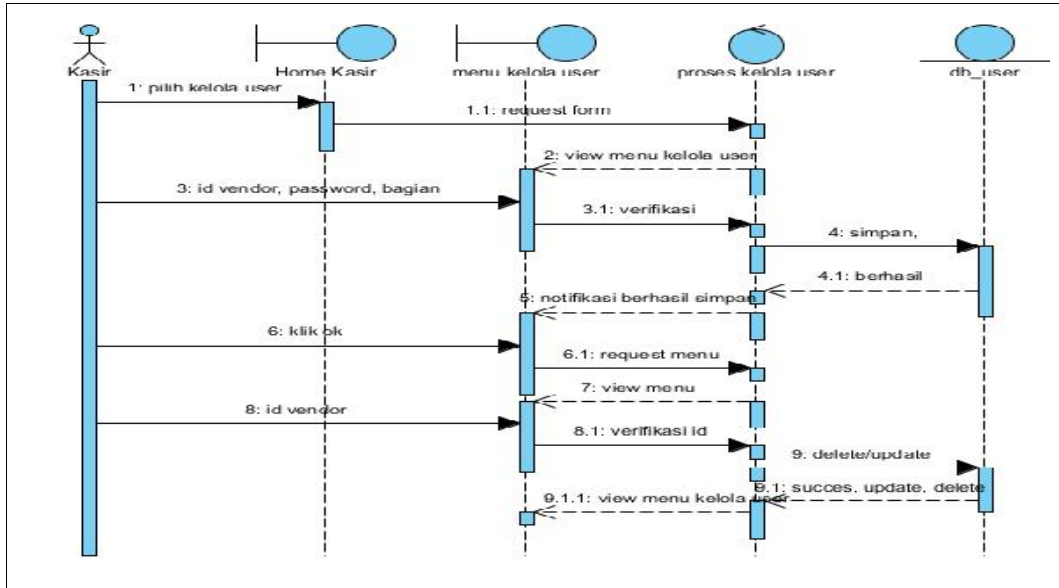
Gambar 6. Sequence Diagram Kelola Pesanan

a. Penjelasan Sequence Diagram Kelola Pesanan

User akan *login*, setelah *login* user akan medapat tampilan menu yang ada, setelah pembeli memilih. User memasukkan pillihan pembeli, disertai nama vendor, jumlah.

Setelah diisi *user* akan menyimpan kedalam *database* secara otomatis system akan mendapat informasi nomor pesanan.

3.9.5 Sequence Diagram Keola User

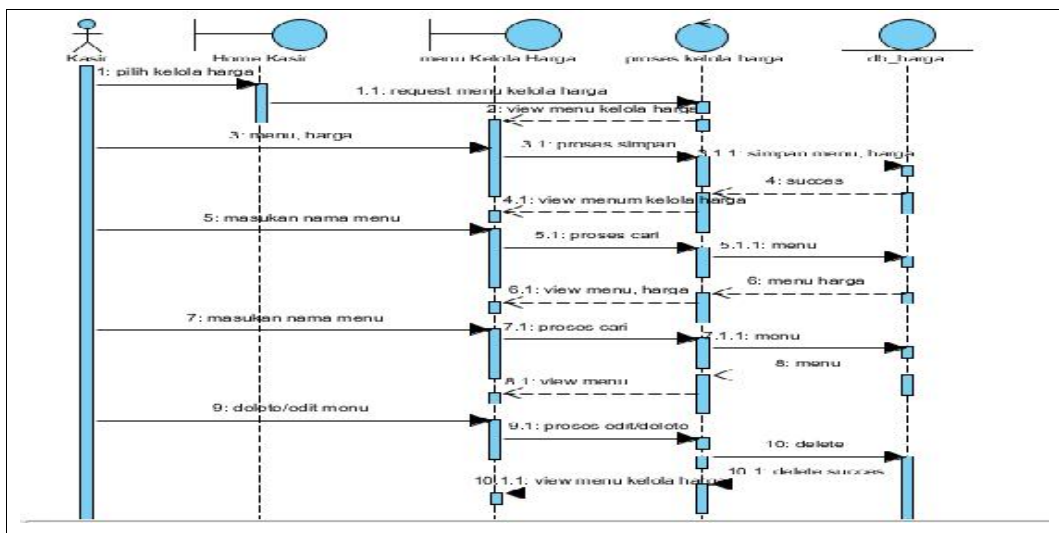


Gambar 11. Sequence Diagram Kelola User

a. Penjelasan Sequence Diagram Kelola User

User akan login, setelah login, pilih menu kelola user, setelah tampil user memasukkan nama vendor, password, dan bagian. Setelah diisi user akan menyimpan data kedalam database.

3.9.6 Sequence Diagram Kelola Harga

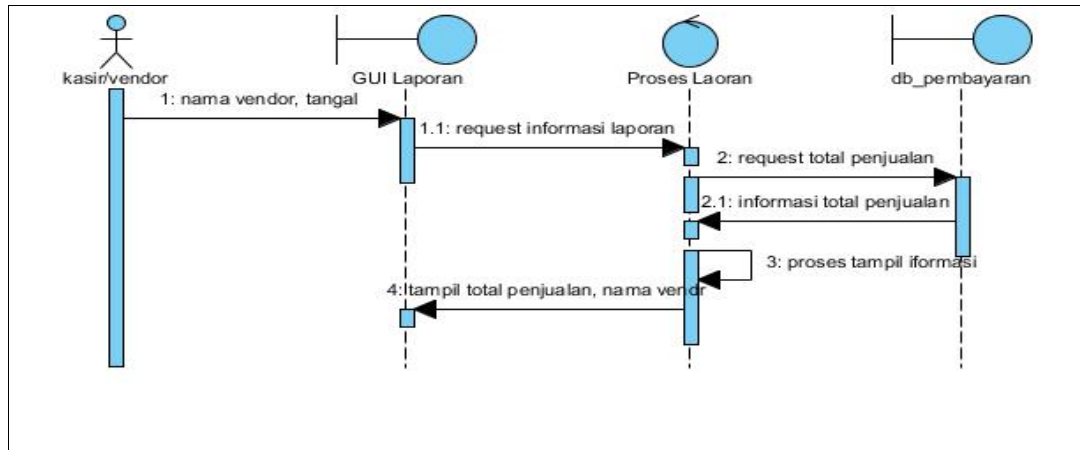


Gambar 12. Sequence Diagram Kelola Harga

a. Penjelasan Sequence Diagram Kelola Harga

User akan login, setelah login, pilih menu kelola harga, user akan memasukkan data berupa nama menu dan harga,.

3.9.6 Sequence Diagram Laporan



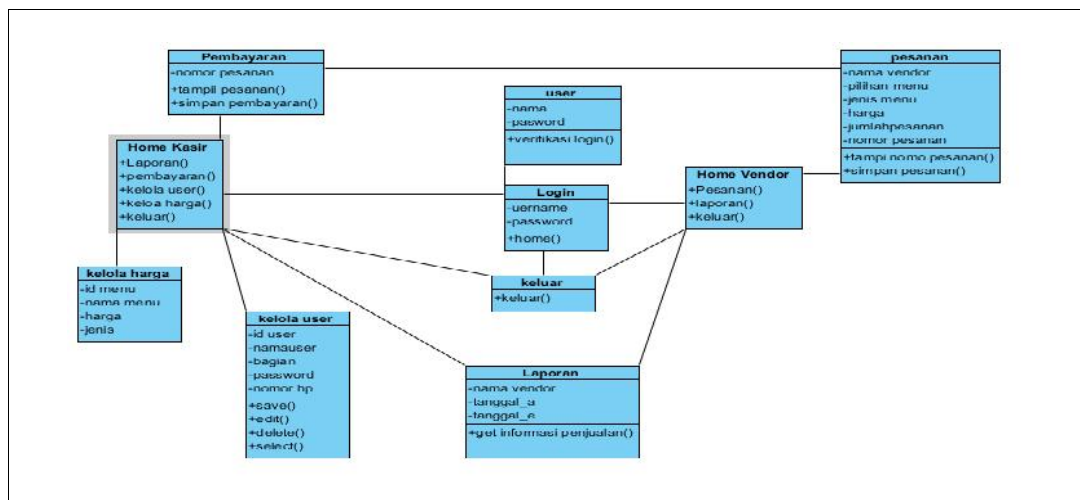
Gambar 13. Sequence Diagram Laporan

a. Penjelasan Sequence Diagram Laporan

User akan login, setelah login, pilih menu Laporan. Sistem akan menampilkan menu laporan. User memasukkan nama vendor disertai tanggal yang diinginkan. Sistem akan mengolah, jika sudah selesai, sistem akan menampilkan laporan berupa total pejualan dan nama vendor

3.10 Class Diagram Android PoS Cloud-based

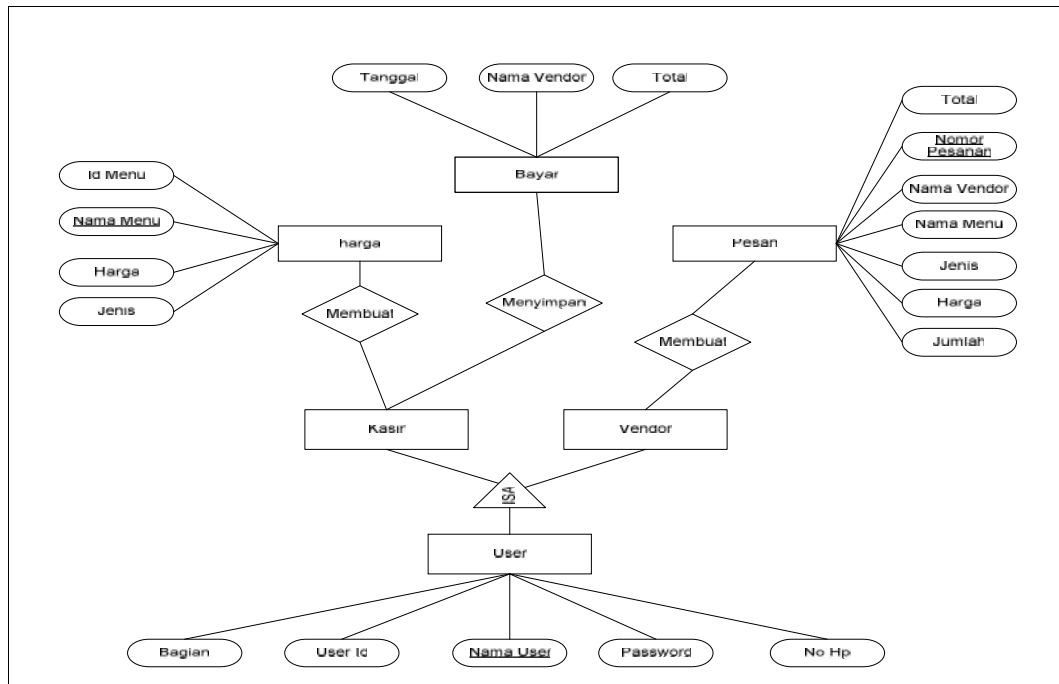
Class Diagram digunakan untuk menampilkan kelas-kelas dan paket-paket dalam system. Berikut ini Class Diagram Android PoS Cloud-based.



Gambar 14. Class Diagram Android PoS Cloud-based

3.11 Perancangan Basis Data

Perancangan basis data yang akan digunakan dalam sistem ini dapat dilihat pada gambar berikut.



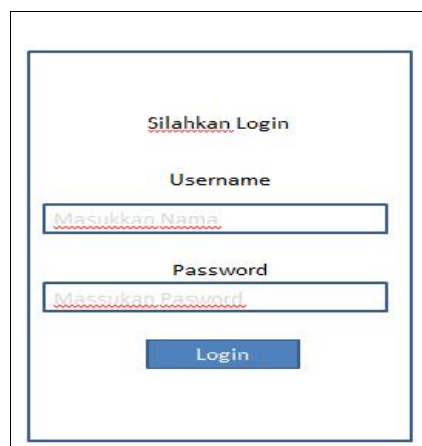
Gambar 15. Perancangn Basis Data

3.12 Perancangan *Interface* Android PoS Cloud-based

Pada bagian ini akan dijelaskan mengenai rancangan antar muka Android PoS Cloud-based.

3.12.1 Antar Muka *login*

Rancangan antar muka untuk *login* dapat dilihat seperti dibawah ini.



Gambar 16. Antar Muka *login*

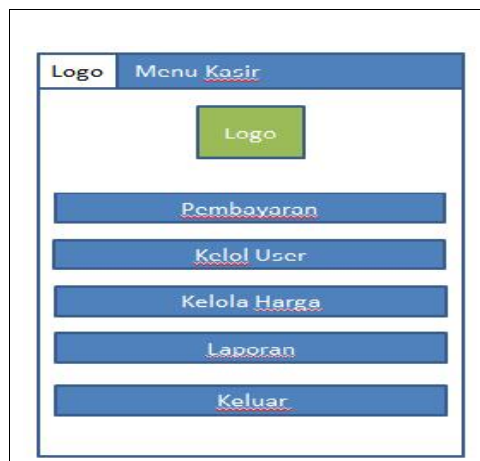
Berikut deskripsi antarmuka halaman *login* :

Table 11. Deskripsi Antar Muka *login*

Id	Jenis	Keterangan	Fungsi
Nama	<i>Input Text</i>	Nama lengkap	diisi dengan <i>username</i>
<i>Password</i>	<i>input password</i>	<i>Password</i>	diisi dengan <i>password</i>
<i>login</i>	<i>Button</i>	<i>login</i>	button untuk <i>login</i>

3.12.2 Antar Muka *Home Kasir*

Rancang antar muka untuk *home kasir* dapat dilihat seperti dibawah ini :



Gambar 17. Antar Muka Menu Kasir

Berikut deskripsi antarmuka halaman *Home Kasir* :

Table 12. Deskripsi Antar Muka Menu kasir

Id	Jenis	Keterangan	Fungsi
kelola_ <i>user</i>	<i>Button</i>	Kelola <i>User</i>	button untuk buat <i>user</i>
kelola_ <i>harga</i>	<i>Button</i>	Kelola Harga	button untuk buat Menu
pembayaran	<i>Button</i>	Pembayaran	button untuk Pembayaran
laporan	<i>Button</i>	Laporan	button untuk Laporan
Keluar	<i>Button</i>	Keluar	button untuk Keluar

3.12.2 Antar Muka *Home Vendor*

Rancang antar muka untuk *home* kasir dapat dilihat seperti dibawah ini :



Gambar 18. Antar Muka *Home Vendor*

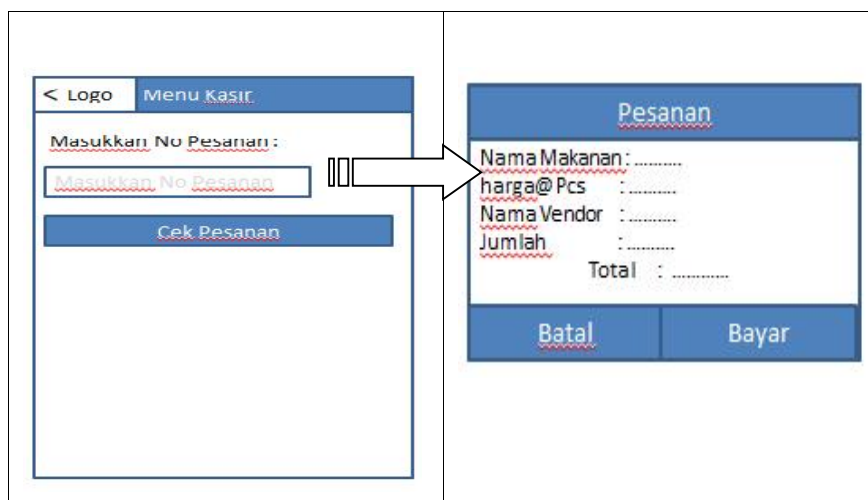
Berikut deskripsi antarmuka halaman *Home Vendor* :

Table 13. Deskripsi Antar Muka Vendor

Id	Jenis	Keterangan	Fungsi
logo	Gambar	Gambar	tampilla
Pilihan menu	Button	Pilihan menu	Menampilkan menu selanjutnya
Laporan	Button	Laporan	Menampilan menu laporan
Keluar	Button	Keluar	keluar dari <i>user</i>

3.12.3 Antar Muka Pembayaran

Rancang antar muka untuk pembayaran dapat dilihat seperti dibawah ini :



Gambar 19. Antar Muka Pembayaran

Berikut deskripsi antar muka halaman Pembayaran :

Table 14. Deskripsi Antar Muka Pembayaran

Id	Jenis	Keterangan	Fungsi
Masukan no pesanan	<i>Input text</i>	Nomor Pesanan	diisi nomor pesanan
Cek	<i>Botton</i>	Cek Pesanan	Menampilkan total pemaaran
Nama makanan	<i>Text view</i>	Nama maanan yg dipesan	Informasi nama makanan yg dipesan
Harga	<i>Text view</i>	Harga per menu	Informasi harga per menu
Nama vendor	<i>Text view</i>	Nama vendor	Informasi nama vendor yang membuat menu
Jumlah	<i>Text view</i>	Jumlah	Informasi Jumlah yang dipesan
Total	<i>Text view</i>	Total	Informasi total yang harus dibayar
Batal	<i>Button</i>	Batal	Membatalkan proses pembayaran
Bayar	<i>Button</i>	Bayar	Menyimpan data pmbayaran ke dalam database

3.12.4 Antar Muka Kelola Harga

Rancang antar muka untuk kelola harga dapat dilihat seperti dibawah ini :



Gambar 20. Antar Muka Kelola Harga

Berikut deskripsi antar muka halaman Kelola Harga :

Table 15. Deskripsi Antar Muka Kelola Harga

Id	Jenis	Keterangan	Fungsi
Pencarian	<i>Button</i>	Pencarian	Mencari menu yang ada di list
New	<i>Button</i>	Buat harga	Menampilan form buat harga
List	<i>Button</i>	List menu	List menu yag ada

3.12.5 Antar Muka Buat Harga

Rancang antar muka untuk buat harga dapat dilihat seperti dibawah ini :



Gambar 21. Antar Muka Buat Harga

Berikut deskripsi antar muka halaman Buat Harga :

Table 16. Deskripsi Antar Muka Buat Harga

Id	Jenis	Keterangan	Fungsi
Menu Id	<i>input text</i>	Menu Id	Diisi id menu
Nama Menu	<i>input text</i>	Nama Menu	diisi dengan nama menu
Jenis	<i>Input text</i>	Jenis	Diisi makanan atau minuman
Harga	<i>Input text</i>	Harga	Diisi dengan harga
Save	<i>Button</i>	Simpan	Menyimpan data ke cloud

3.12.6 Antar Muka Lihat Harga

Rancang antar muka untuk lihat harga dapat dilihat seperti dibawah ini :



Gambar 22. Antar Muka Lihat Harga

Berikut deskripsi antar muka halaman Lihat Harga :

Table 17. Deskripsi Antar Muka Lihat Harga

Id	Jenis	Keterangan	Fungsi
menu	<i>Button</i>	List	List menu untuk melihat detail menu

3.12.7 Antar Muka Kelola *user*

Rancang antar muka untuk kelola *user* dapat dilihat seperti dibawah ini :



Gambar 23. Antar Muka Kelola *User*

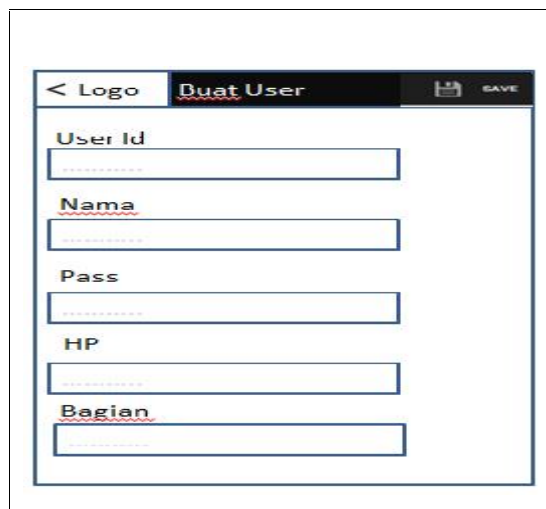
Berikut deskripsi antar muka halaman Kelola *User* :

Table 18. Deskripsi Antar Muka Kelola *User*

Id	Jenis	Keterangan	Fungsi
Pencarian	<i>Button</i>	Pencarian	Mencari <i>user</i> yang ada di list
New	<i>Button</i>	Buat <i>user</i>	Menampilkan form buat <i>user</i>
List	<i>Button</i>	List <i>user</i>	List <i>user</i> yang ada

3.12.8 Antar Muka Buat User

Rancang antar muka untuk buat *user* dapat dilihat seperti dibawah ini :



Gambar 24. Antar Muka Buat User

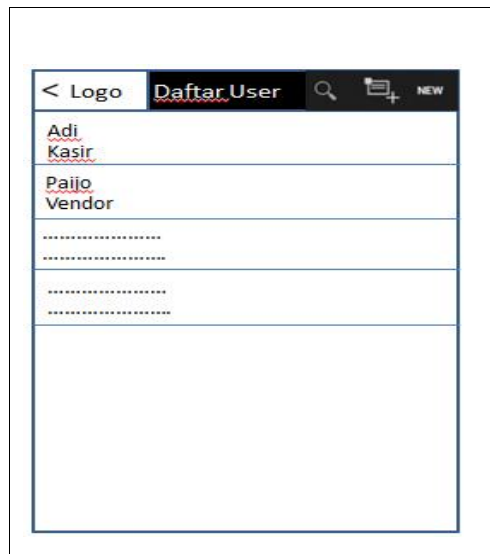
Berikut deskripsi antar muka halaman Kelola *User* :

Table 19. Deskripsi Antar Muka Buat User

Id	Jenis	Keterangan	Fungsi
<i>User Id</i>	<i>input text</i>	<i>User Id</i>	Diisi id <i>user</i>
<i>Nama user</i>	<i>input text</i>	<i>Nama User</i>	diisi dengan nama <i>user</i>
<i>Pass</i>	<i>Input text</i>	<i>Pass</i>	Diisi password
<i>Hp</i>	<i>Input text</i>	<i>Hp</i>	Diisi nomor hp
<i>Bagian</i>	<i>Input text</i>	<i>Bagian</i>	Diisi bagian (kasir/vendor)
<i>Save</i>	<i>Button</i>	<i>Simpan</i>	Menyimpan data ke cloud

3.12.9 Antar Muka Lihat *User*

Rancang antar muka untuk lihat *user* dapat dilihat seperti dibawah ini :



Gambar 25. Antar Muka Lihat *User*

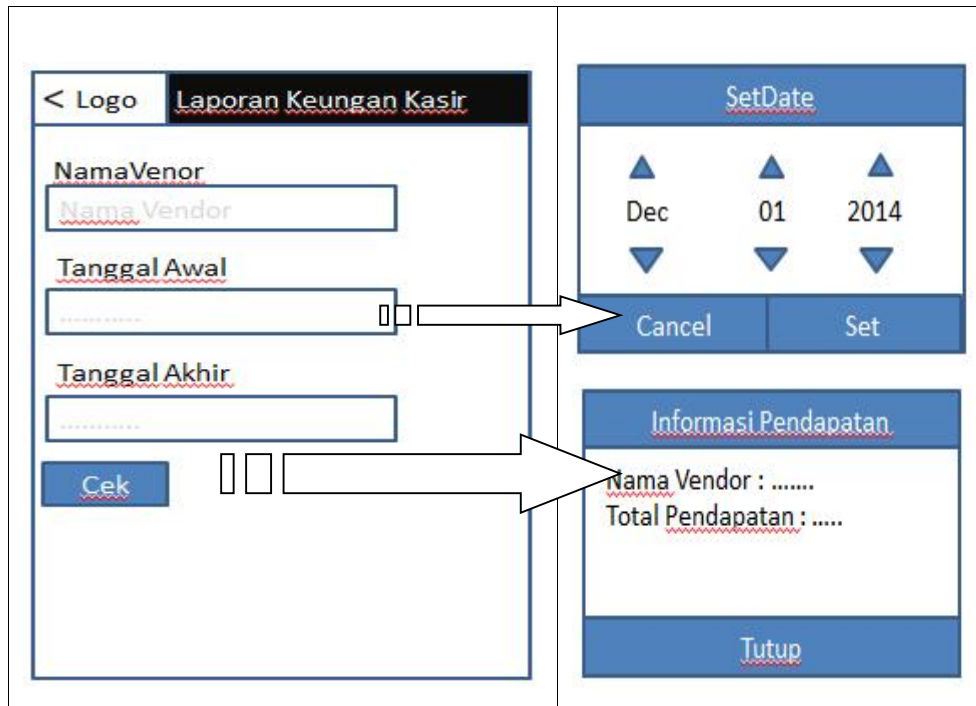
Berikut deskripsi antar muka halaman Lihat *User* :

Table 20. Deskripsi Antar Muka Lihat *User*

Id	Jenis	Keterangan	Fungsi
Nama <i>User</i>	<i>Button</i>	List	List menu untuk melihat detail <i>User</i>

3.12.9 Antar Muka Laporan

Rancang antar muka untuk Laporan dapat dilihat seperti dibawah ini :



Gambar 26. Antar Muka Laporan

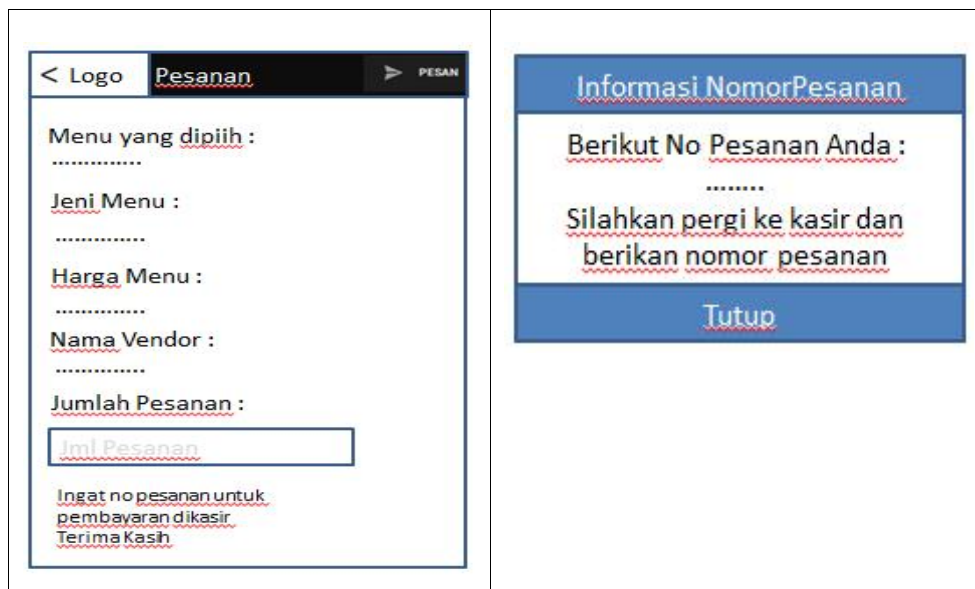
Berikut deskripsi antar muka halaman Laporan :

Table 21. Deskripsi Antar Muka Laporan

Id	Jenis	Keterangan	Fungsi
Nama vendor	<i>Input text</i>	Id Vendor	parameter untuk menampilkan laporan
Tanggal awal	<i>input text</i>	Tanggal awal	parameter untuk menampilkan laporan
Tanggal Akhir	<i>input text</i>	Tanggal Akhir	parameter untuk menampilkan laporan
Total_pendapatan	<i>Text View</i>	Total Pendapatan	Informasi total Pendpatan
Cek	<i>Button</i>	Cek	Meminta informasi pendapatan
Tutup	<i>Button</i>	Tutup	Menutup informasi pendapatan

3.12.10 Antar Muka Pesanan

Rancang antar muka untuk Pesanan dapat dilihat seperti dibawah ini :



Gambar 27. Antar Muka Pesanan

Berikut deskripsi antar muka halaman pesanan :

Table 22. Deskripsi Antar Muka Pesanan

Id	Jenis	Keterangan	Fungsi
Menu	<i>Text view</i>	Menu yang dipilih	Menu yang dipilih oleh <i>user</i>
Jenis Menu	<i>Text view</i>	Jenis menu	Jenis menu (makanan/minuman)
Harga Menu	<i>Text view</i>	Harga menu	Harga per menu
Nama Vendor	<i>Text View</i>	Nama Vendor	Nama Vendor pembuat makanan
Jumlah pesanan	<i>Input text</i>	Jumlah pesanan	Jumlah pesanan yang diinginkan
Pesan	<i>Button</i>	Pesan	Mengirim data ke database
Nomor pesanan	<i>Text view</i>	Nomor pesanan	Angka nomor pesanan
Tutup	<i>Button</i>	tutup	Menutup informasi nomor pesanan

BAB 4

HASIL DAN PEMBAHASAN

4.1 Prosedur Aplikasi Android PoS Cloud-based

Aplikasi ini dibuat agar dapat memudahkan vendor/tenan dan pengurus kasir dalam mengolah data. Harapanya dengan adanya aplikasi ini tidak akan ada lagi kehilangan data atau kesalahan data antara vendor dan pengelola.

4.2 Perancangan Aplikasi Android PoS Cloud-based

Aplikasi ini dibangun dengan eclipse dan android SDK sebagai *tool building* sedangkan untuk media penyimpanan bisa divirtualkan dengan MySQL dengan batuan *xampp*, sebagai *virtualisasi* dari *cloud* yang sebenarnya. Setelah siap menjadi APK baru diimplementasikan dengan layanan *cloud computing*. Perancangan aplikasi ini menggunakan *visual paradigma* untuk membuat *Use Case*, *Sequence Diagram*, dan *Class Diagram*.

4.3 Impementasi Database

Database dibuat menggunakan bantuan *xampp/phpmyadmin*. Berikut daftar tabel *db_pos* untuk aplikasi Android PoS Cloud-based.

1. Tabel “*user*” berisi tentang informasi *user* berikut strutur tabel *user*

```
CREATE TABLE IF NOT EXISTS `user` (  
  `id` int(50) NOT NULL AUTO_INCREMENT,  
  `nama` varchar(50) NOT NULL,  
  `pass` varchar(50) NOT NULL,  
  `bagian` varchar(50) NOT NULL,  
  `hp` varchar(50) NOT NULL,  
  PRIMARY KEY (`nama`),  
  UNIQUE KEY `id` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1  
AUTO_INCREMENT=6 ;
```

2. Tabel “bayar“ berisi tentang data pembayaran yang dilakukan berikut struktur tabel bayar

```
CREATE TABLE IF NOT EXISTS `bayar` (  
  
    `idvendor` varchar(50) NOT NULL,  
  
    `total` double NOT NULL,  
  
    `date` date NOT NULL  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

3. Tabel “harga” berisi tentang data menu dan harga makanan maupun minuman yang dijual yang dilakukan berikut struktur tabel harga

```
CREATE TABLE IF NOT EXISTS `harga` (  
  
    `id` int(50) NOT NULL AUTO_INCREMENT,  
  
    `nama` varchar(50) NOT NULL,  
  
    `jenis` varchar(50) NOT NULL,  
  
    `harga` varchar(50) NOT NULL,  
  
    `vendor` varchar(100) NOT NULL,  
  
    PRIMARY KEY (`id`),  
  
    UNIQUE KEY `id` (`id`)  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=latin1  
    AUTO_INCREMENT=39 ;
```

4. Tabel “pesan” berisi tentang pesanana makanan maupun minuman yang dijual berikut struktur tabel pesan

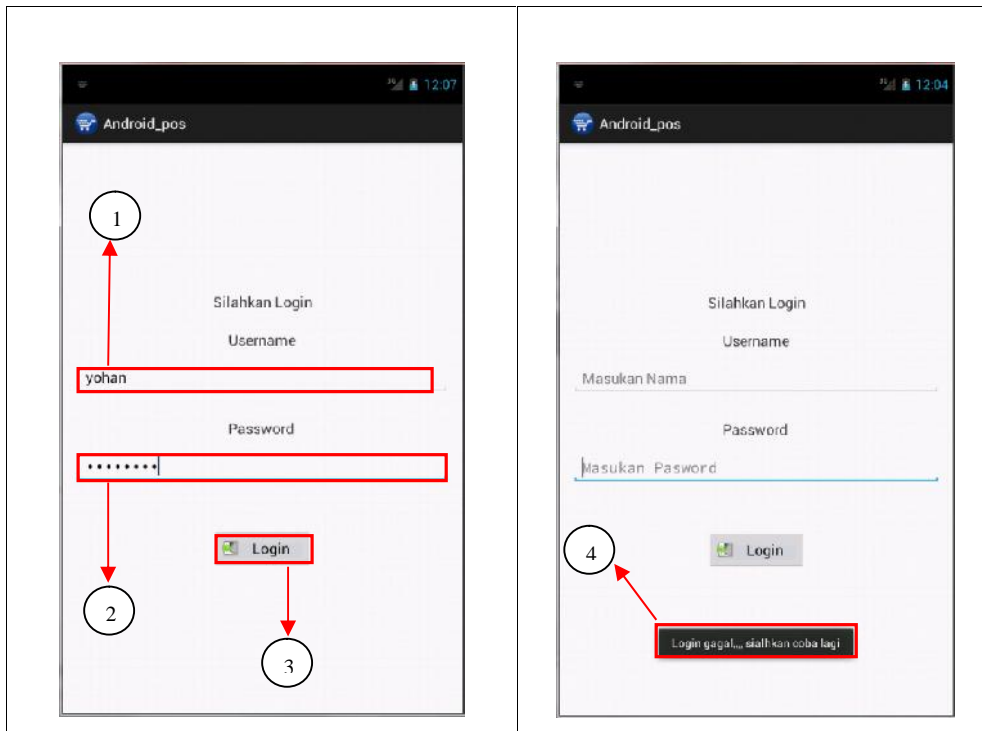
```
CREATE TABLE IF NOT EXISTS `pesan` (  
  
  `user` varchar(50) NOT NULL,  
  
  `nama` varchar(50) NOT NULL,  
  
  `jenis` varchar(50) NOT NULL,  
  
  `harga` int(50) NOT NULL,  
  
  `jumlah` int(50) NOT NULL,  
  
  `nopes` int(100) NOT NULL,  
  
  `total` double NOT NULL,  
  
  PRIMARY KEY (`nopes`)  
  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.4 Impementasi Antar Muka (*Interface*)

Tahapan ini bertujuan untuk menjelaskan secara singkat penggunaan Aplikasi Android PoS Cloud-based, berikut penjelasanya.

4.4.1 Impementasi *Frame login*

login merupakan syarat bagi *user* (Kasir/Vendor) untuk dapat menggunakan fitur-fitur yang terdapat pada aplikasi Android PoS Cloud-based, berikut gambaran *frame login* pada Android PoS Cloud-based.



Gambar 28. Frame login

Keterangan gambar :

1. *Username.*

Username merupakan identitas dari kasir maupun vendor yang sudah terdaftar dan mendapatkan akses untuk menggunakan Aplikasi Android PoS Cloud-based berdasarkan akses yang didapat. Diisi dengan nama *user* dalam bentuk *string*

2. *Password*

Password merupakan syarat kedua setelah *username*. *Password* berguna untuk keamanan. *Password* menampilkan karakter yang diketikkan oleh *user* dalam bentuk (●●●●●●)

3. *login Button*

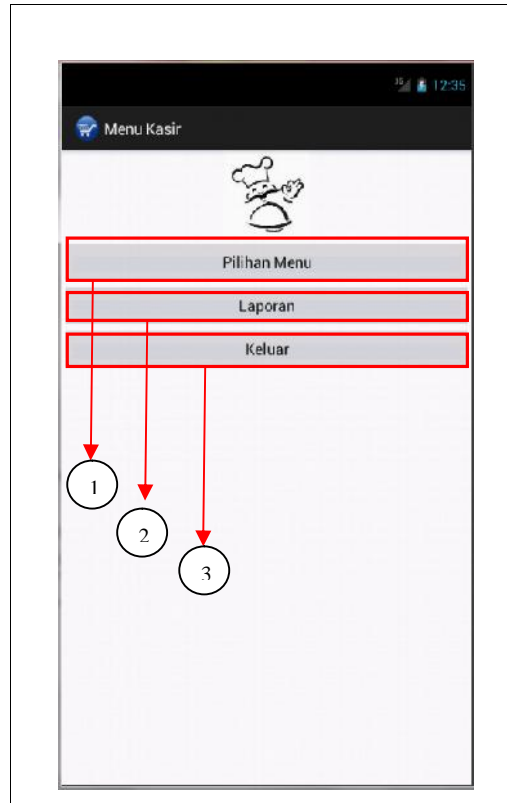
Button login berfungsi untuk memulai proses verifikasi ke database dengan parameter *username* dan *password* yang telah diisi.

4. *Notification*

Notifikasi setelah *button login* diklik. Tampilan *notifikasi* bergantung dari hasil verifikasi system, jika *username* dan *password* tidak sesuai maka akan muncul notifikasi “ *login gagal silahkan coba lagi*” jika berhasil akan muncul notifikasi “.....(nama *user*) anda berhasil *login*”.

4.4.2 Implementasi *Frame Home Vendor*

Home vendor merupakan tampilan awal yang mempunyai beberapa fitur guna membantu vendor. Fitur yang ada antara lain (Pilihan Menu, Laporan, Keluar). Berikut gambaran *frame Home* vendor pada Android PoS Cloud-based.



Gambar 29. *Frame Home Vendor*

Keterangan :

1. *Button* Pilihan Menu

Button Pilih Menu merukan *button* yang akan menampilkan daftar menu yang dijual oleh vendor.

2. *Button* Laporan

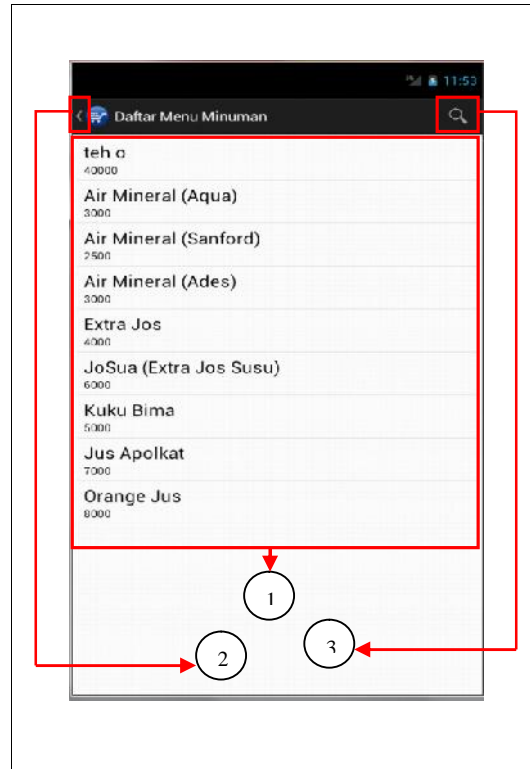
Button Laporan merukan *button* yang akan menampilkan menu Laoran untuk vendor.

3. *Button* Keluar

Button Keluar merukan *button* yang memberikan akses keluar dari aplikasi, jika di klik maka *user* akan kembali ke menu *login*.

4.4.3 Impementasi *Frame* Daftar Menu

Daftar Menu merupakan list makanan dan minuman yan dijual oleh vendor. Berikut gambaran *frame* daftar menu pada Android PoS Cloud-based.



Gambar 30. Daftar Menu

Keterangan :

1. List Makanan dan Minuman

List ini berisi makanan maupun minuman yang disediakan oleh vendor, dalam list tersebut dapat dilihat nama menu da harga

2. *Button Home* (“<”)

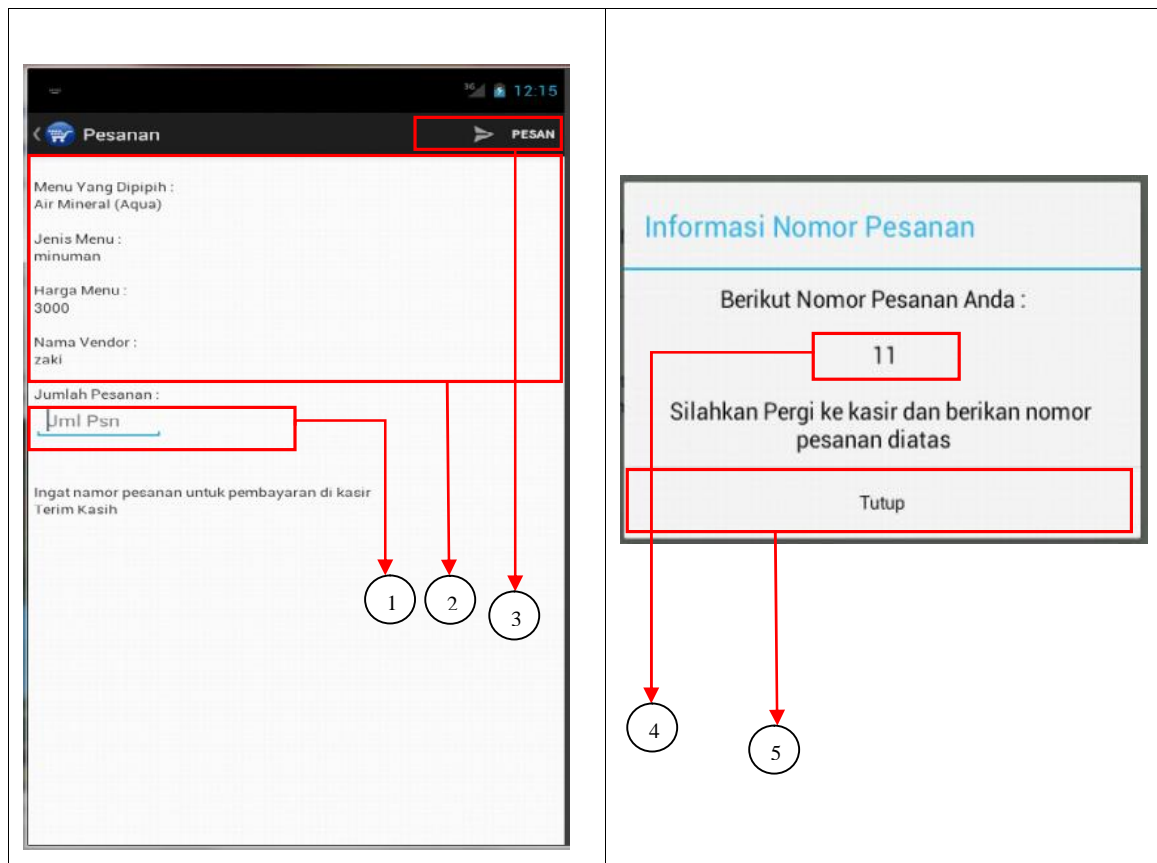
Button Home digunakan untuk kembali ke frame sebelumnya.

3. Menu Pencarian

Menu ini dapat digunakan untuk mencari makanan maupun minuman yang ada dilist berdasarkan nama makanan dan harga

4.4.4 Impementasi *Frame* Pesanan

Frame Pesanan adalah *frame* yang berisi detail menu yang akan dipesan serta button dan inputan jumlah yang digunakan untuk proses pemesanan. Berikut gambaran *frame* Pesanan pada Android PoS Cloud-based.



Gambar 311. *Frame Pesanan*

Keterangan :

1. *Input Text* Jumlah Pesanan

Inputan ini diisi dengan jumlah item yang dibeli, dari inputan yang nantinya dijadikan perhitungan untuk pembayaran.

2. *Detail* menu

Detail menu berisi keterangan secara detail menu dari *list* yang ada, detail makanan ini diisi secara otomatis oleh sistem

3. *Button* pesan

Button ini digunakan untuk mengirim data menu yang dipesan ke *cloud* agar mendapatkan nomor pesanan untuk proses pembayaran

4. Nomor pesanan

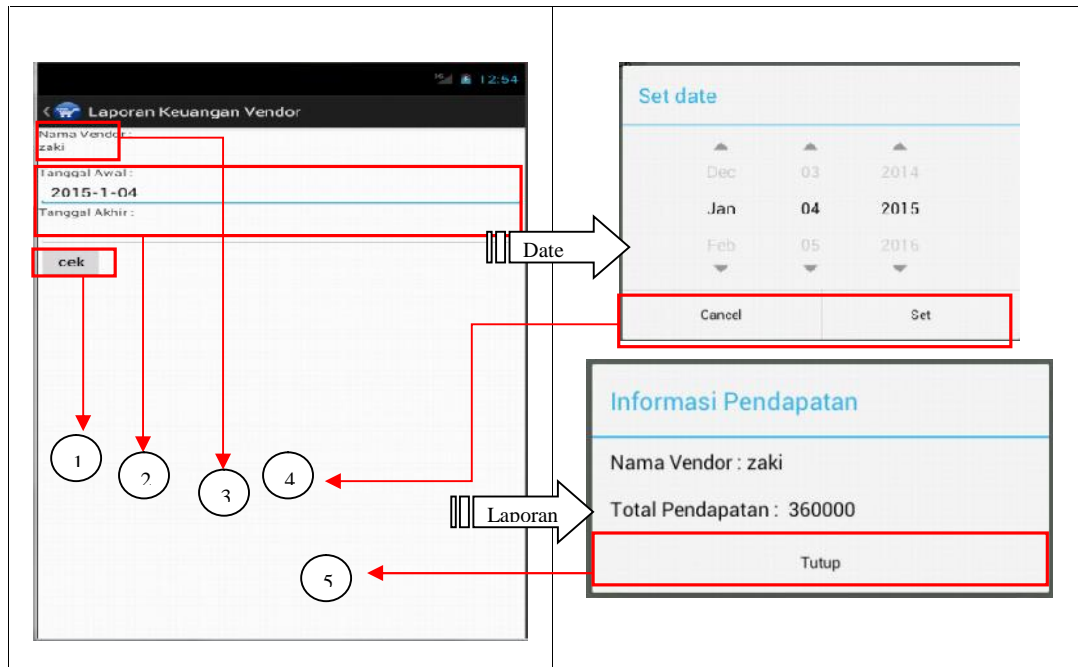
Nomor pesanan akan didapat setelah proses pesanan dilakukan

5. *Button* tutup

Button ini berfungsi untuk menutup informasi nomor pesanan.

4.4.5 Impementasi *Frame* Laporan Vendor

Frame Laporan Vendor adalah *frame* digunakan untuk mengetahui jumlah pendapatan berdasarkan tanggal. Berikut gambaran *frame* Laporan Vendor pada Android PoS Cloud-based.



Gambar 322. *Frame* Laporan Vendor

Keterangan :

1. *Button* Cek

Button ini digunakan untuk menampilkan informasi penjualan berdasarkan *parameter* nama vendor dan tanggal.

2. Tanggal

Inputan ini akan digunakan menjadi *parameter* untuk *request* data ke dalam *cloud/database*. *Format* tanggal adalah bulan, hari dan tahun.

3. Nama vendor

Sama halnya dengan tanggal inputan ini digunakan menjadi *parameter*, bedanya untuk nama vendor *user* tidak harus mengisi, nama vendor akan terisi otomatis berdasarkan *login* awal.

4. *Button* pada *date*

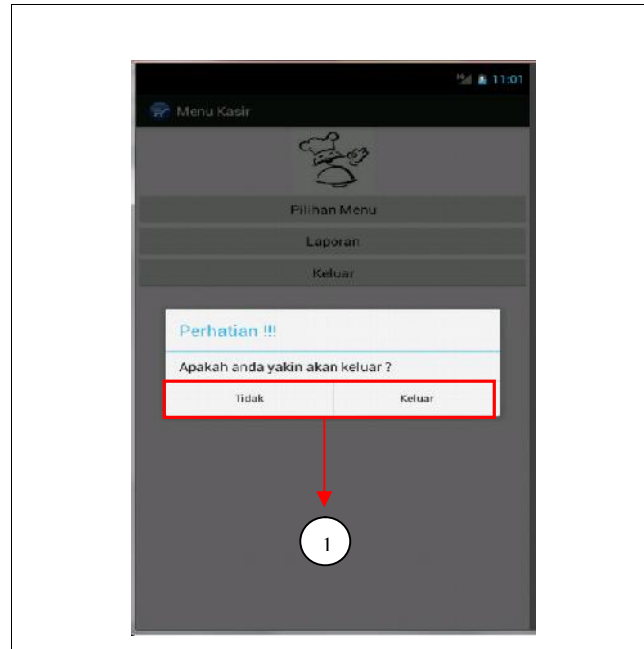
Button tersebut diguakan untuk memasukan tanggal kedalam inputan tanggal, maupun membatalkan inputan.

5. *Button* tutup

Button ini diguakan untuk menutup laporan.

4.4.6 Impementasi *Button* Keluar

Button ini digunakan untuk keluar dari aplikasi ke menu *login*. Berikut gambaran *button* keluar pada Android PoS Cloud-based.



Gambar 333. Button Keluar

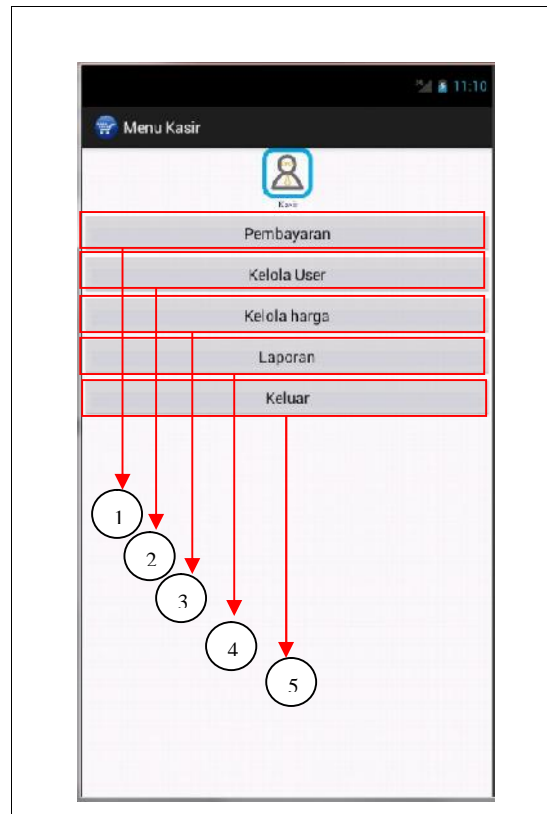
Keterangan :

1. *Button* keluar

Button keluar adalah buton yang digunakan untuk keluar dari aplikasi dan kembali ke menu *login*, sebelum dieksekusi system akan meenampilkan pesan dan menyediakan pilihan keluar atau tidak. Jika keluar maka akan menuju ke *frame login*.

4.4.7 Impementasi *Frame Home* Kasir

Home Kasir merupakan tampilan awal yang mempunyai beberapa fitur guna membantu kasir. *Fitur* yang ada antara lain (Pembayaran, Kelola *User*, Kelola Harga, Laporan, dan Keluar). Berikut gambaran *frame Home* vendor pada Android PoS Cloud-based



Gambar 344. Frame Home Kasir

Keterangan :

1. *Button* Pembayaran

Button ini akan menampilkan frame pembayaran yang akan digunakan untuk proses pembayaran.

2. *Button* Kelola user

Button Kelola user digunakan untuk menampilkan *frame* CRUD terhadap user

3. *Button* Kelola harga

Button Kelola harga digunakan untuk menampilkan *frame* CRUF terhadap harga

4. *Button* laporan

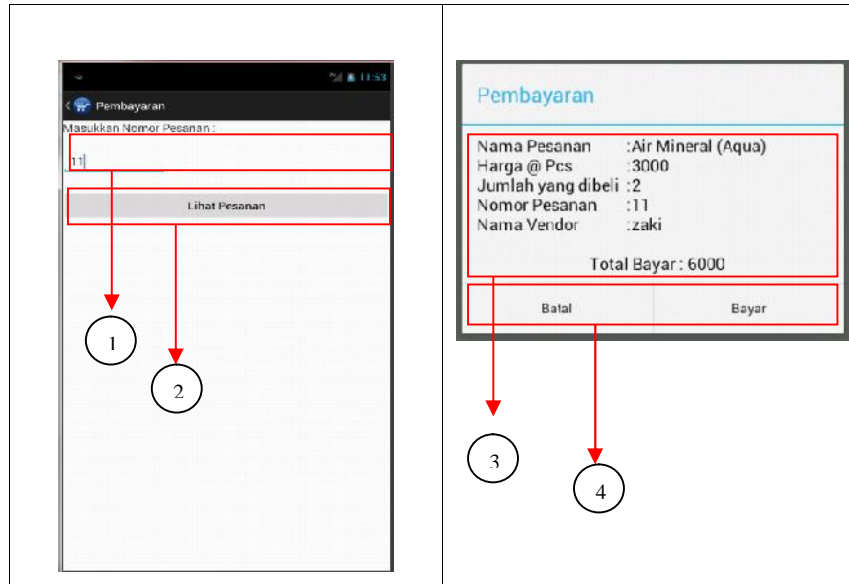
Button Laporan merukan *button* yang akan menampilkan menu Laporan untuk vendor.

5. *Button* Keluar

Button Keluar merukan *button* yang memberikan akses keluar dari aplikasi, jika di klik maka *user* akan kembali ke menu *login*.

4.4.8 Implementasi *Frame* Pembayaran

Pembayaran merupakan tampilan dimana kasir dapat melakukan transaksi pembayaran dan menyimpan data kedalam *databases/cloud*. Berikut gambaran *frame* *Home* vendor pada Android PoS Cloud-based.



Gambar 355. *Frame* Pembayaran

Keterangan :

1. *Input* nomor pesanan

Inputan ini diisi nomor pesanan. Nomor pesanan tersebut akan digunakan menjadi parameter untuk menampilkan informasi detail pembayaran.

2. *Button* lihat pesanan

Button ini digunakan untuk lihat jumlah pembayaran dari inputan nomor pesanan.

3. Detail Pembayaran

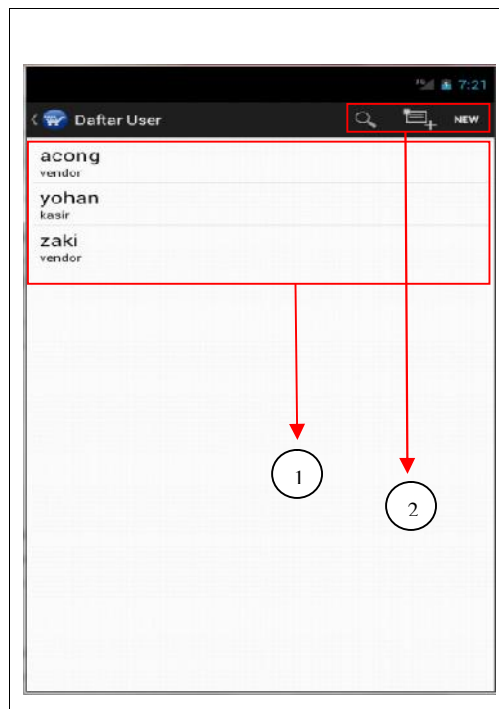
Detail pembayaran merupakan detail pesanan yang telah dipesan oleh pembeli disertai total pembayara

4. *Button* Bayar

Button bayar dan batal adalah *botton* yang digunakan untuk meyimpan data pembayaran kedalam *database*.

4.4.9 Impementasi Kelola *User* (Daftar *User*).

Frame Kelola *user* merupakan fitur yang diberikan kepada kasir untuk digunakan dalam bidang CRUD terhadap *user*. Sedangkan untuk Kelola *user* (Daftar *User*) digunakan untuk menampilkan semua *user* beserta bagianya. Berikut gambaran *frame* Kelola *user* pada Android PoS Cloud-based.



Gambar 366. Frame Kelola *User* (Daftar *User*)

Keterangan :

1. Daftar *user*

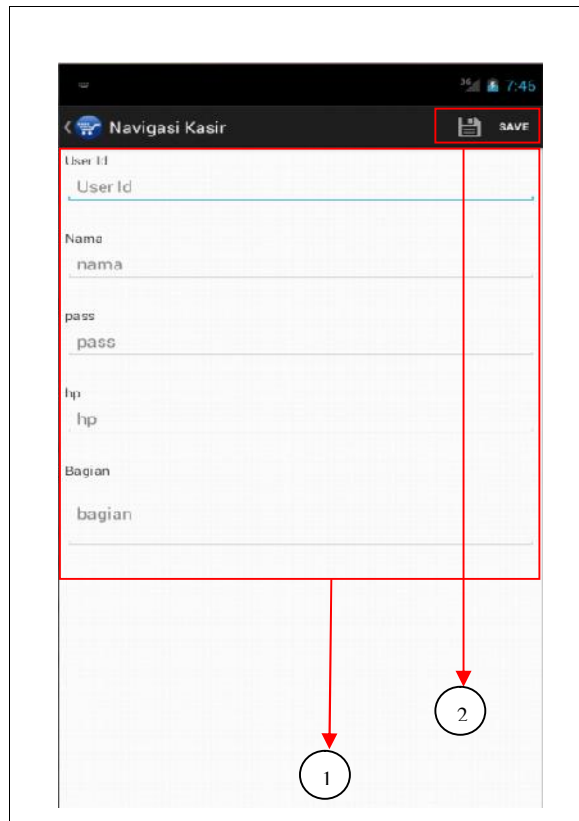
Daftar *user* berisi nama-nama *user* dan bagian-bagian dari masing-masing *user* tersebut

2. *Button new*

Button new digunakan untuk menampilkan *frame* pembuatan *user* baru.

4.4.10 Impementasi Kelola *User* (Buat *User*).

Frame Kelola *user* merupakan *fitur* yang diberikan kepada kasir untuk digunakan dalam bidang CRUD terhadap *user*. Sedangkan untuk Kelola *user* (Buat *User*) digunakan untuk membuat *user* baru. Berikut gambaran *frame* Kelola *user* (buat *user*) pada Android PoS Cloud-based.



Gambar 377. Frame Kelola User (Buat User)

Keterangan :

1. Inputan data *user*

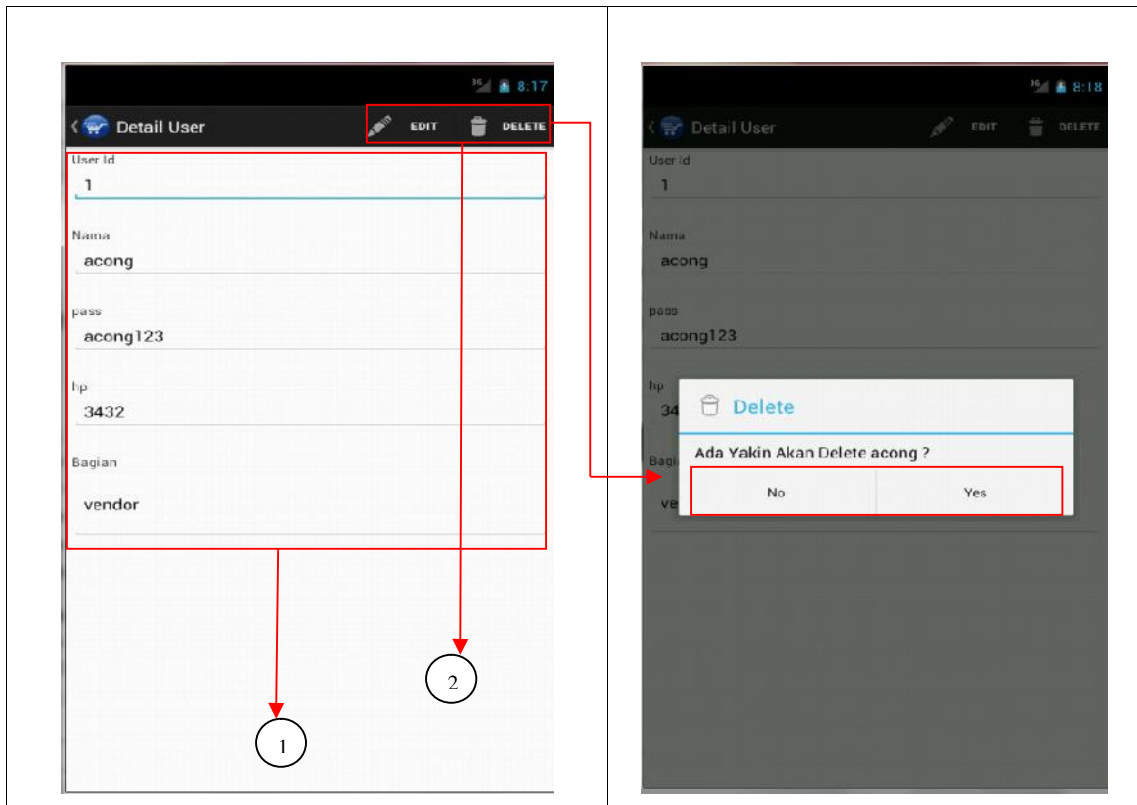
Inputan ini diisi data *user* yang akan menggunakan aplikasi ini, disini juga *user* di berikan hak akses berdasarkan bagian

2. *Button save*

Button ini digunakan untuk menyimpan data *user* kedalam *cloud/database*.

4.4.11 Impementasi Kelola User (Edit dan Delete User).

Frame Kelola user merupakan fitur yang diberikan kepada kasir untuk diguakan dalam bidang CRUD terhadap *user*. Sedangkan untuk Kelola *user (Edit dan Delete User)* digunakan untuk *edit* maupun *delete user*. Berikut gambaran *frame Kelola user (edit dan delete user)* pada Android PoS Cloud-based.



Gambar 388. Frame Kelola User (Edit & Delete User)

Keterangan :

1. Informasi detail *user*

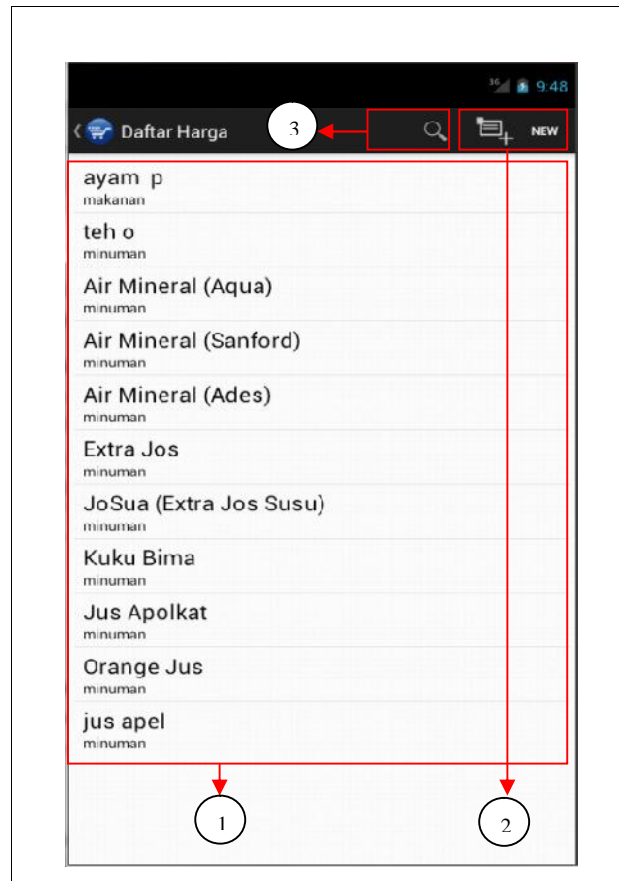
Dalam informasi detail ini jika kasir menginginkan edit maka kasir tinggal pilih menu *edit* maka detail *user* secara otomatis dapat diedit.

2. *Button delete*

Button ini digunakan untuk menghapus *user* dari *cloud/database*. Setelah dihapus maka secara otomatis *user* tidak dapat mengakses aplikasi ini.

4.4.12 Implementasi Kelola Harga (Daftar Harga).

Frame Kelola user merupakan fitur yang diberikan kepada kasir untuk digunakan dalam bidang CRUD terhadap *user*. Sedangkan untuk Kelola *user* (Daftar *User*) digunakan untuk menampilkan semua *user* beserta bagianya. Berikut gambaran *frame Kelola user* pada Android PoS Cloud-based.



Gambar 39. Frame Kelola Harga (Daftar Harga)

Keterangan :

1. *List* daftar Makanan

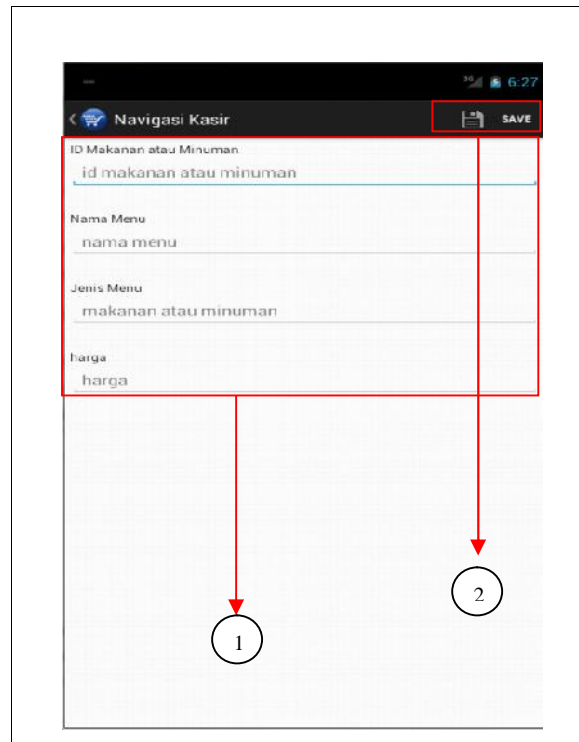
List daftar makanan menampilkan semua jenis menu makanan dan minuman yang dijual oleh semua vendor. Untuk memudahkan pencarian tinggal menekan button pencarian seperti yang ditunjukkan no. 3 dalam gambar 40. Pada pencarian *user* dapat memasukan nama makanan ataupun melakukan pencarian berdasarkan jenis.

2. *Button new*

Button new digunakan untuk menampilkan *frame* pembuatan *user* baru.

4.4.13 Impementasi Kelola Harga (Buat Harga).

Frame Kelola Harga merupakan fitur yang diberikan kepada kasir untuk digunakan dalam bidang CRUD terhadap harga dan menu. Sedangkan untuk Kelola *user* (Buat Harga) digunakan untuk membuat menu dan harga baru. Berikut gambaran *frame* Kelola harga (buat harga) pada Android PoS Cloud-based.



Gambar 390. Kelola Harga(Buat Harga)

Keterangan :

1. *Input* data harga

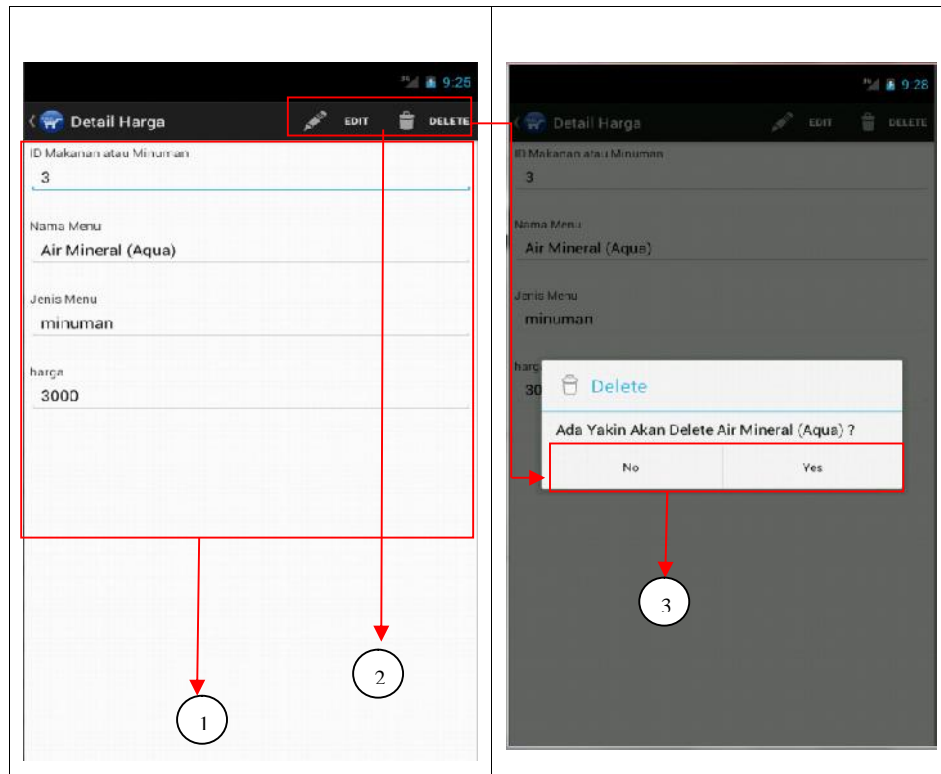
Input ini diisi dengan data harga dan menu yang dijual atau disediakan oleh vendor

2. *Button save*

Button ini digunakan untuk menyimpan data harga dan menu ke dalam *cloud/database*.

4.4.14 Implementasi Kelola Harga (*Edit dan Delete* Harga).

Frame Kelola *user* merupakan fitur yang diberikan kepada kasir untuk digunakan dalam bidang CRUD terhadap harga. Sedangkan untuk Kelola harga (*Edit dan Delete* harga) digunakan untuk *edit* maupun *delete* harga. Berikut gambaran *frame* Kelola harga (*edit dan delete* harga) pada Android PoS Cloud-based.



Gambar 401. Kelola Harga(Edit Dan Delete Harga)

Keterangan :

1. Informasi Detail Harga

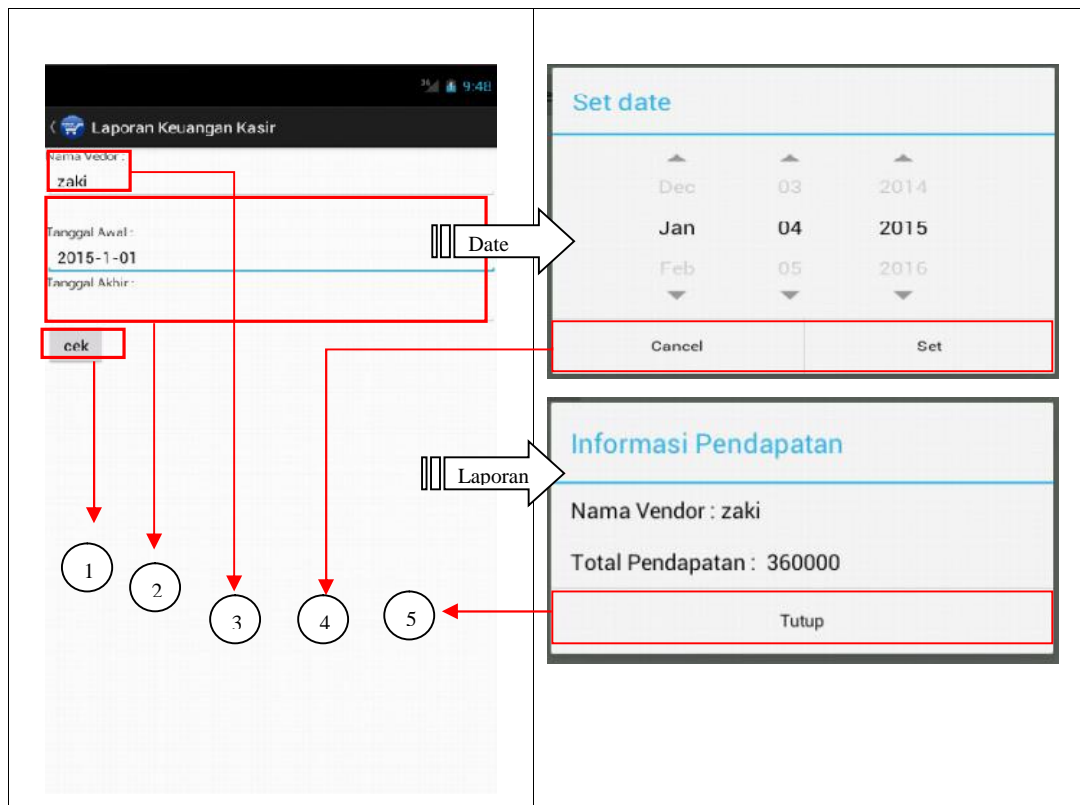
Dalam informasi detail harga ini adalah fitur yang diberikan kepada kasir untuk mengedit harga maupun menu yang ada.

2. *Button delete*

Button ini digunakan untuk menghapus harga dan menu yang ada. Jika *button* di tekan maka akan muncul notifikasi seperti yang terlihat nomor 3 pada gambar 42. Notifikasi tersebut memberikan kesempatan untuk *user* meyakinkan pilihan. Jika ditekan “yes” maka harga dan menu akan terhapus.

4.4.15 Impementasi Laporan Kasir.

Frame Laporan Kasir adalah *frame* digunakan untuk mengetahui jumlah pendapatan berdasarkan tanggal dan nama vendor yang diinputkan oleh kasir, sedikit berbeda dengan laporan vendor. Pada laporan vendor nama vendor tidak diinputkan secara manual tapi sudah diberikan oleh system dan tidak dapat diedit. Berikut gambaran *frame* Laporan Kasir pada Android PoS Cloud-based.



Gambar 412. Laporan Kasir

Keterangan :

1. *Button cek*

Button ini digunakan untuk menampilkan informasi penjualan berdasarkan parameter nama vendor dan tanggal.

2. *Tanggal*

Inputan ini berupa tanggal, ada dua macam yang harus di masukkan yaitu tanggal awal dan tanggal akhir. Format tanggal adalah bulan, hari, tahun.

3. *Nama vendor*

Seperti yang telah dijelaskan nama vendor pada laporan kasri harus dimasukkan secara manual. Inputan ini akan digunakan menjadi parameter sama halnya dengan tanggal.

4. *Button set dan cancel pada date.*

Button set digunakan untuk memeasukkan tanggal kedalam inputan, *button cancel* digunakan untuk membatalkan inputan

5. *Button Tutup*

Button ini diguakan untuk menutup laporan.

4.5 Implementasi Cloud

Implementasi *Cloud* adalah implementasi file-file PHP yang di tempatkan dalam *cloud* agar aplikasi dapat mengakses data dari *database*. Berikut isi file PHP yang disimpan dalam *cloud*

Table 23. Tabel File PHP

No	Nama File	Extensi	Fungsi
1	Connection	PHP	Koneksi Ke <i>Database</i>
2	Bayar	PHP	Menyimpan hasil pembayaran
3	cek_pesanan	PHP	Cek nomor pesanan
4	Cek	PHP	Validasi <i>user</i>
5	delete_harga	PHP	Delete harga
6	delete_user	PHP	Delete user
7	Laporan	PHP	Menampilkan informasi laporan
8	Pesan	PHP	Menyimpan pesanan dan menampilkan informasi nomor pesanan
9	select_all	PHP	Menampilkan seluruh user
10	select_harga	PHP	Menampilkan seluruh Harga dan menu
11	select_makanan	PHP	Menampilkan Informasi menu berdasarkan vendor
12	submit_harga	PHP	Menyimpan harga dan menu baru
13	submit_user	PHP	Menyimpan <i>user</i> baru
14	update_harga	PHP	update data harga dan menu
15	update_user	PHP	update data user

4.6 Pengujian Android PoS Cloud-based

4.6.1 Tujuan Pengujian

Pengujian sistem bertujuan untuk menemukan kesalahan yang mungkin masih terdapat dalam sistem dan untuk mengetahui apakah program yang dibuat telah sesuai dengan tujuan yang telah ditetapkan.

4.6.2 Strategi Pengujian

Strategi pengujian pada Android PoS Cloud-based menggunakan metode *Black Box* yaitu berfokus pada kebutuhan deskripsi fungsional guna untuk mengetahui jalannya sistem secara lengkap.

4.6.3 Hasil Pengujian

Berikut hasil pengujian aplikasi PoS Cloud-based :

Table 24. Tabel Pengujian

No	Proses yang diuji	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan	Penguji
1	login	Mengosongkan semua isi data <i>login</i> dan menekan button <i>login</i>	Nama : - Password :-	Sistem menolak akses <i>login</i> dan menampilkan pesan " <i>User</i> tidak boleh kosong"	Sesuai Harapan	Valid	Zaki A Agha
		Mengosongkan nama <i>user</i> haya mengisi nama password menekan button <i>login</i>	Nama : - Password : yo*****	Sistem menolak akses <i>login</i> dan menampilkan pesan " <i>User</i> tidak boleh kosong"	Sesuai Harapan	Valid	Zaki A Agha
		Mengosongkan nama password haya mengisi nama <i>user</i> menekan button <i>login</i>	Nama : yohan Password :-	Sistem menolak akses <i>login</i> dan menampilkan pesan "Password tidak boleh kosong"	Sesuai Harapan	Valid	Zaki A Agha
		Mengisi data dengan data yang salah	Nama : yohan Password : acinfo	Sistem menolak akses <i>login</i> dan menampilkan pesan " <i>login</i> Gagal Silahkan Coba Lagi"	Sesuai Harapan	Valid	Zaki A Agha
		Mengisi data dengan data yang benar	Nama : yohan Password : yohan	tampilke menu home kasir maupun vendor tergantung akses	Sesuai Harapan	Valid	Zaki A Agha
2	Pesanan	kosongkan jumlah pesanan menekan button pesan	jumlah pesanan : -	Sistem menolak memproses pesanan dan menampilkan pesan "Jumlah tidak boleh kosong"	Sesuai Harapan	Valid	Zaki A Agha
		mengisi jumlah pesanan	jumlah pesanan : 5	menampilkan informasi nomor pesanan	Sesuai Harapan	Valid	Zaki A Agha

3	Laporan Vendor	Mengkosongkan tanggal menekan button cek	Tanggal awal : - Tanggal awal : -	Sistem menolak memproses pesanan dan menampilkan pesan "Tanggal tidak boleh kosong"	Sesuai Harapan	Valid	Zaki A Agha
		mengisi tanggal awal dan mengkosongkan tanggal akhir dan mekan button cek	Tanggal awal :- Tanggal akhir : 2015-01-5	Sistem menolak memproses pesanan dan menampilkan pesan "Tanggal awal tidak boleh kosog"	Sesuai Harapan	Valid	Zaki A Agha
		mengkosongkan tanggal akhir dan mengisi tanggal awal dan mekan button cek	Tanggal awal : 2015-01-5 Tanggal akhir :-	Sistem menolak memproses pesanan dan menampilkan pesan "Tanggal akhir tidak boleh kosog"	Sesuai Harapan	Valid	Zaki A Agha
		mengisi tanggal awal dan akhir dan mekan button cek	Tanggal awal : 2015-01-5 Tanggal awal : 2015-01-5	Informasi penjualan	Sesuai Harapan	Valid	Zaki A Agha
4	Pembayaran	mengkosongkan nomor pesanan, dan menekan button lihat pesanan	nomor pesanan : -	Sistem menolak request informasi pesanan dan memberikan indormasi "nomor pesanan tidak boleh kosong"	Sesuai Harapan	Valid	Zaki A Agha
		Mengisi nomor pesanan, dan menekan button lihat pesanan	nomor pesanan : 5	tampil informasi detail pesanan, serta pilihan untuk membayar atau membatalkan	Sesuai Harapan	Valid	Zaki A Agha
		menekan button bayar pada informasi detail pesanan	tekan button bayar	Sistem melakukan penyimpanan kedatabase,jika berhasil akan memunculkan informasi "Data berhasil disimpan" jika gagal akan menampilkan pesan "data gagal disimpan".	Sesuai Harapan	Valid	Zaki A Agha

5	Kelola User	mengkosongkan semua data dan menekan button save	User Id : - Nama : - Pass : - Hp: - Bagian : -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data User Id dan menekan button save	User Id : 1 Nama : - Pass : - Hp: - Bagian : -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data User Id dan Nama kemudian menekan button save	User Id : 1 Nama : Zaki Pass : - Hp: - Bagian : -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data User Id, Nama dan Pass kemudian menekan button save	User Id : 1 Nama : Zaki Pass : zaki\$1a Hp: - Bagian : -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data User Id, Nama, Hp dan Pass kemudian menekan button save	User Id : 1 Nama : Zaki Pass : zaki\$1a Hp: 08123344 Bagian : -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data Nama, Hp, bagian dan Pass kemudian menekan button save	User Id : - Nama : Zaki Pass : zaki\$1a Hp: 08123344 Bagian : vendor	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha

5	Kelola User	hanya mengisi data <i>User</i> Id, Hp, bagian dan Pass kemudian menekan button save	<i>User</i> Id : 1 Nama : - Pass : zaki\$1a Hp: 08123344 Bagian : vendor	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data <i>User</i> Id, nama, Hp, dan bagian kemudian menekan button save	<i>User</i> Id : 1 Nama : zaki Pass : - Hp: 08123344 Bagian : vendor	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data <i>User</i> Id, nama, pass, dan bagian kemudian menekan button save	<i>User</i> Id : 1 Nama : zaki Pass : zaki\$1a Hp: - Bagian : vendor	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		mengisi semua data dengan lengkap kemudian menekan button save	<i>User</i> Id : 1 Nama : zaki Pass : zaki\$1a Hp: 0812345 Bagian : vendor	Memberikan informasi penyimpanan, jika tersimpan nama <i>user</i> ada di dalam list. Jika gagal maka memberikan pesan "data gagal disimpan"	Sesuai Harapan	Valid	Zaki A Agha
		Delete salah satu <i>user</i> .	melihat detail <i>user</i> , dan setelah itu meneekan button delete. Setelah muncul notifikasi klik button yes	<i>User</i> terhapus dari daftar list	Sesuai Harapan	Valid	Zaki A Agha
6	Kelola harga	mengkosongkan semua data dan menekan button save	Id makanan atau Minuman : - Nama Menu: - Jenis Menu : - Harga: -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha

6	Kelola harga	hanya mengisi data Id Makanan atau Minuman dan menekan button save	Id makanan atau Minuman : 1 Nama Menu: - Jenis Menu : - Harga: -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data Id Makanan atau Minuman dan Nama Menu kemudian menekan button save	Id makanan atau Minuman : 1 Nama Menu: Ayam Penyet Jenis Menu : - Harga: -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data Id Makanan atau Minuman , Nama Menu dan Jenis Menu kemudian menekan button save	Id makanan atau Minuman : 1 Nama Menu: Ayam Penyet Jenis Menu : Makanan Harga: -	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data Nama Menu, Jenis Menu, dan Harga kemudian menekan button save	Id makanan atau Minuman : - Nama Menu: Ayam Penyet Jenis Menu : Makanan Harga: 12000	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		hanya mengisi data Id Makanan atau Minuman, Jenis Menu dan, Harga kemudian menekan button save	Id makanan atau Minuman : 1 Nama Menu: - Jenis Menu : Makanan Harga: 12000	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha

6	Kelola harga	hanya mengisi data Id Makanan atau Minuman, Nama Menu dan, Harga kemudian menekan button save	Id makanan atau Minuman : 1 Nama Menu: Ayam Penyet Jenis Menu : - Harga: 12000	Sistem menolak menyimpan data dan memberikan informasi "data harus terisi semua"	Sesuai Harapan	Valid	Zaki A Agha
		mengisi semua data dengan lengkap kemudian menekan button save	Id makanan atau Minuman : 1 Nama Menu: Ayam Penyet Jenis Menu : Makanan Harga: 12000	Memberikan informasi penyimpanan, jika tersimpan nama <i>user</i> ada di dalam list. Jika gagal maka memberikan pesan "data gagal disimpan"	Sesuai Harapan	Valid	Zaki A Agha
		Delete salah satu harga / menu.	melihat detail harga/menu, dan setelah itu menekan button delete. Setelah muncul notifikasi klik button yes	harga atau menu terhapus dari daftar list	Sesuai Harapan	Valid	Zaki A Agha
7	Laporan Kasir	Mengkosongkan tanggal menekan button cek	Nama vendor : - Tanggal awal : - Tanggal akhir : -	Sistem menolak memproses pesanan dan menampilkan pesan "Data Harus terisi"	Sesuai Harapan	Valid	Zaki A Agha
		Mengisi nama vendor dan menekan button cek	Nama vendor : zaki Tanggal awal : - Tanggal akhir : -	Sistem menolak memproses pesanan dan menampilkan pesan "Data Harus terisi"	Sesuai Harapan	Valid	Zaki A Agha
		mengisi tanggal awal, nama vendor dan mengkosongkan tanggal akhir dan menekankan button cek	Nama vendor: zaki Tanggal awal : 2015-01-5 Tanggal akhir :-	Sistem menolak memproses pesanan dan menampilkan pesan "Tanggal awal tidak boleh kosong"	Sesuai Harapan	Valid	Zaki A Agha

7	Laporan Kasir	mengkosongkan tanggal awal dan mengisi tanggal akhir, nama vendor dan menekan button cek	Nama vendor: zaki Tanggal awal :- Tanggal akhir : 2015-01-5	Sistem menolak memproses pesanan dan menampilkan pesan "Tanggal akhir tidak boleh kosog"	Sesuai Harapan	Valid	Zaki A Agha
		menkosongkan nama vendor dan mengisi tanggal awal dan akhir, selanjutnya menekan button cek	Nama Vendor: - Tanggal awal : 2015-01-5 Tanggal awal : 2015-01-5	Sistem menolak memproses pesanan dan menampilkan pesan "Tanggal akhir tidak boleh kosog"	Sesuai Harapan	Valid	Zaki A Agha
		Mengisi semua data secara lengkap	Nama Vendor: zaki Tanggal awal : 2015-01-5 Tanggal awal : 2015-01-5	Sistem menolak memproses pesanan dan menampilkan pesan "Tanggal akhir tidak boleh kosog"	Sesuai Harapan	Valid	Zaki A Agha

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari Tugas Akhir Andoid PoS Cloud-based sebagai berikut :

1. Aplikasi dapat berjalan di *platform Operating Sistem* Android
2. Aplikasi dapat menyimpan data di *cloud*.

5.2 Saran

Berikut saran untuk pengembang aplikasi Android PoS Cloud-based :

1. Pada fitur pembayaran ditambahkan proses *print* otomatis sehingga dapat terkoneksi dengan printer melalui media *bluethoot* atau *usb* guna pencetakan kwitansi
2. Menambahkan fitur NFC card agar pembayaran tidak dilakukan secara *cash*
3. Membuat *web* untuk pemilik vendor maupun pihak pegelola guna laporan lebih rinci.

DAFTAR PUSTAKA

- Nazruddin Safaat H, 2012, *Pemrograman Aplikasi Mobile Smartphone dan Tablet Berbasis Android*. Penerbit Informatika Bandung.
- Wei-Meng Lee, 2012, *Beginning Android™ Application Development*, John Wiley & Sons, Inc.
- Ryan A.P Armanda, 2010, *Implementasi Teknologi Cloud Computing Menggunakan Cloudsim Untuk Implementasi Tik Hijau*, ST. Disertai Universitas Indonesia, Jakarta
- Novita, Edin Surdi Djatikusuma, M.Si, 2013, *Perancangan Sistem Informasi Poin Of Sale(POS) pada PD Tokyo*, STMIK GI MDP
- Abednego Dinarim Siregar, David Michael Simanjuntak, 2014, *e-Kantin Politeknik Negeri batam*. Politeknik Negeri Batam.
- Sri Dharwiyanti, 2003, *Pengantar Unified Modeling Language (UML)*, Kuliah Umum Ilmu Komputer.com, diakses pada tanggal 12 Oktober 2014
- Abdul Kadir, 2005, *Dasar Pemrograman Java 2*. Penerbit Andy Yogyakarta,
- Stendy B. sakur, 2010, *PHP 5 Pemograman Berorientasi Object*. Penerbit Andi,
- JSON Developer, 2014, *Pengenalan JSON*, Website : <http://www.json.org/json-id.html>, diakses tanggal 6 Januari 2014

LAMPIRAN

LAMPIRAN A.

USER REQUIREMENT

No	Requitrement	Request By User	Sign By User Requirement	
1	Saplikasi dapat mengganti sitem manual yang ada	Syahrial	1	
2	aplikasi dapat diterapkan tanpa merubah sistem yang ada	Syahrial		2
3	aplikasi dapat menyimpan data secara terintegrasi	Syahrial	3	
4	Aplikasi dapat menangani proses pemesanan secara mandiri	Syahrial		4
5	Aplikasi dapat menangani proses pembayaran	Syahrial	5	
6	Aplikasi dapat menampilkan laporan penjualan berdasarkan tanggal	Syahrial		6
7	laporan dapat diakses dimanapun dengan fasilitas internet	Syahrial	7	
8	Kasir dapat melalukan proses pembuatan <i>user</i> baru	Syahrial		8
9	Kasir dapat melakukan proses pembuatan menu dan harga baru	Syahrial	9	
10	Aplikasi menyediakan fitur Edit, delete <i>user</i> maupun menu	Syahrial		10

Disetujui Oleh

Yohan Nor (3311211030)
Pengembang Software

LAMPIRAN B.

INFORMASI HOSTING

Aplikasi ini menggunakan hosting sebagai media penyimpanan data agar dapat diakses dari internet. Berikut informasi mengenai hosting yang digunakan dalam aplikasi Android PoS Cloud-based.

Nama Hosting	:	www.ganesa.comli.com
Kapasitas Memori	:	1,5 Gb
Status	:	Free Hosting
Nama Data Base	:	db_pos
Transfer rate	:	100 Mbps
Server Name	:	server23.000webhost.com
Apache ver.	:	2.2.19 (Unix)
PHP version	:	5.2
My SQL ver	:	5.1

LAMPIRAN C.

MENU YANG SUDAH TERSIMPAN DALAM CLOUD

Berikut daftar menu kantin politeknik negeri batam yang sudah dimasukkan kedalam cloud. Menu dibagi menjadi dua berdasarkan jenis yaitu minuman dan makanan

Daftar Menu Minuman :

No	Nama	Harga
1	Air Mineral (Sanford)	2000
2	Sirup Melon	3000
3	Sirup Apel	3000
4	Sirup Mangga	3000
5	Sirup Sirsak	3000
6	Sirup Jeruk	3000
7	Sirup Leci	3000
8	Sirup Jambu	3000
9	Sirup Melon + Susu	5000
10	Sirup Mangga + Susu	5000
11	Sirup Sirsak + Susu	5000
12	Sirup Jeruk +Susu	5000
13	Sirup Leci + Susu	5000
14	Sirup Jambu + Susu	5000
15	Kopi	4000
16	Kopi +Susu	5000
17	Susu	4000
18	Extra Jos	4000
19	Kuku Bima	4000
20	Extra Jos + Susu	5000
21	Kuku Bima + Susu	5000

Daftar Menu Makanan :

No	Nama	Harga
1	Ayam Penyet	13000
2	Ayam Penyet Kalasan	14000
3	Ayam Cabe Ijo	14000
4	Ayam Bumbu Kacang	14000
5	Ayam Penyet	13000
6	Ayam BBQ	15000
7	Mie Goreng + Nasi	10000
8	Mie Rebus + Nasi	10000
9	Mie Setan Lv. 1,2,3	11000
10	Mie Setan Keju Lv. 1,2,3	12000
11	Mie Spaghetti	10000
12	Mie Spaghetti Keju	11000
13	Kwietiau Goreng	10000
14	Kwietiau Siram	11000
15	Bakso Sapi/Ikan	8000

Nama dan harga kedua jenis menu diatas sudah sesuai dengan nama dan harga pada kantin Politeknik Negeri Batam tanggal 14 Januri 2015.