



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.

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Penulis berkedudukan di bagian *Software Development Department, IT Division, Corporate IT&IS*, dan ditugaskan bersama tim yang mengembangkan proyek ini sebagai *programmer* aplikasi *mobile*. Tim ini terdiri dari Marco Hudaya yang menangani sistem Host-to-Host dengan Bank Mandiri, Wirawan Utomo yang menangani platform web, dan Bapak Paksi Buwono yang menangani integrasi AR. Koordinasi dilakukan dengan Bapak Arnold Rompas, *Software Development Manager*, sebagai supervisor dan pembimbing lapangan sekaligus sebagai project manager.

3.2 Tugas yang Dilakukan

Secara umum tugas yang dilakukan penulis dalam kesempatan kerja magang selama kurang lebih dua bulan adalah:

- Mempelajari konsep pengembangan aplikasi mobile menggunakan html 5 dan javascript dengan *tools* DevExtreme,
- Merancang dan mengembangkan WCF OData Service sebagai basis aplikasi *mobile* yang akan dikembangkan ,
- Mengembangkan aplikasi *mobile* menggunakan *tools* DevExtreme.

- Implementasi di dalam *development environment*.

3.3 Uraian Pelaksanaan Kerja Magang

3.3.1 Proses Pelaksanaan

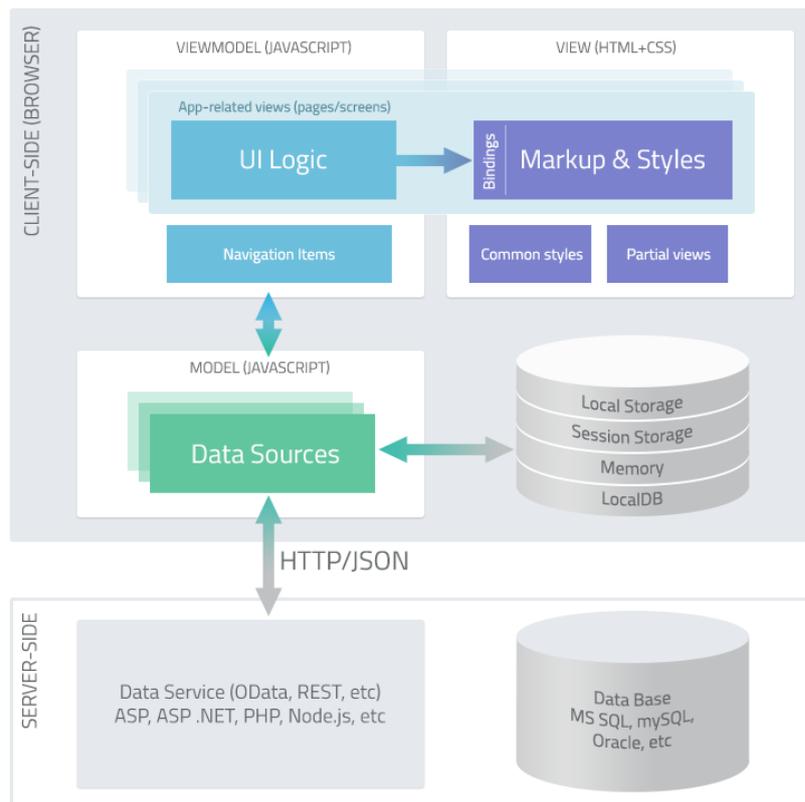
Ada beberapa hal yang dilakukan saat melaksanakan kerja magang, diantaranya mempelajari konsep pengembangan aplikasi mobile menggunakan *tools* DevExtreme, pengembangan WCF OData Service sebagai feeder aplikasi mobile, pengembangan aplikasi mobile dalam *development environment*, dan melakukan testing internal.

3.3.1.1. Konsep Pengembangan Aplikasi menggunakan *tools* DevExtreme

Pada pengembangan aplikasi pada *tools* DevExtreme Bahasa pemrograman yang digunakan adalah HTML5 dan *javascript*. DevExtreme menggunakan konsep *Model-ViewModel-View* yang terdiri dari:

- *Model* yang direpresentasikan oleh *javascript object* yang menyediakan data atau menerima data dari penyimpanan yang bisa terdapat dalam *local storage* maupun *remote server*.
- *ViewModel* yang direpresentasikan oleh *file .js* yang menggunakan *javascript* dari setiap view yang digunakan untuk memanipulasi data yang akan ditampilkan

- *View* yang direpresentasikan oleh *file* .dxview yang menggunakan HTML5 untuk menampilkan data yang sudah diolah.

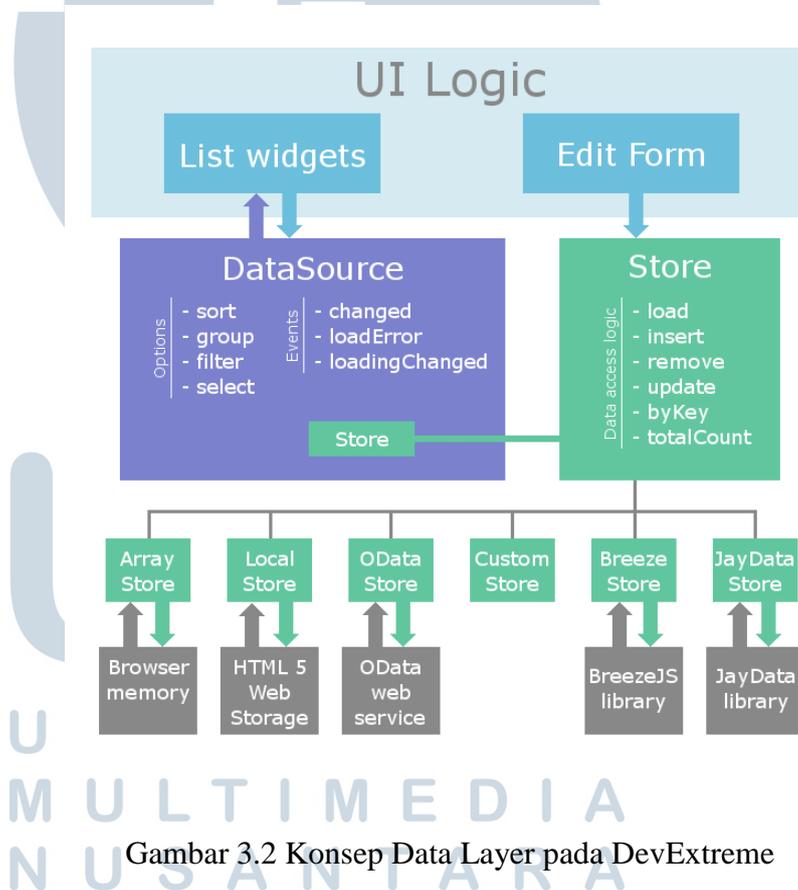


Gambar 3.1 Konsep Model-ViewModel-View pada DevExtreme

DevExtreme memiliki *data layer*, yang terdiri dari berbagai komponen yang membuat dimungkinkannya untuk membaca atau menulis data. Komponen utama dari *data layer* adalah *DataSource* dan *Stores*

DataSource digunakan untuk mempermudah untuk menggunakan dan memanipulasi data seperti melakukan *sorting*, *grouping*, *filtering*, dll tanpa mengubah data itu sebenarnya.

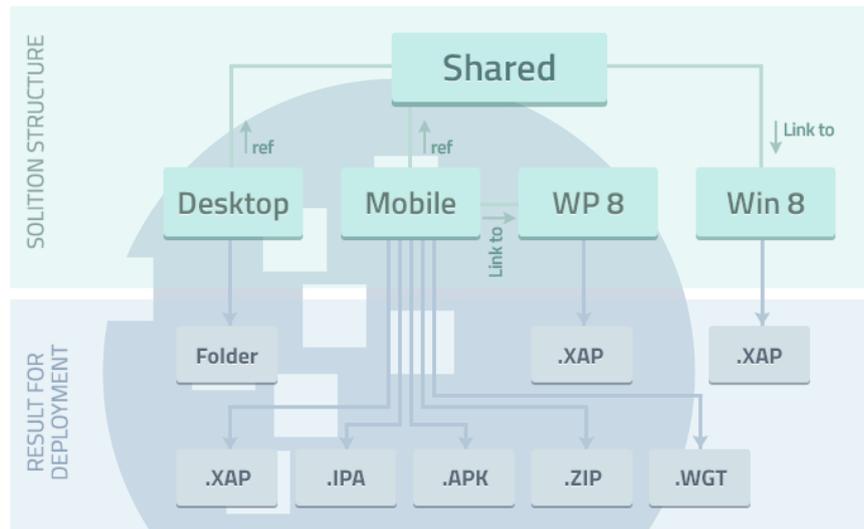
Store adalah interface akses data universal yang diterima dalam DevExtreme . Data yang diterima dari sumber disimpan dalam store sebelum dimanipulasi di *view* melalui *DataSource*.



Gambar 3.2 Konsep Data Layer pada DevExtreme

DevExtreme mendukung *multi-channel* yang terdiri dari *channel web*, *mobile*, *windows phone 8*, dan *windows 8*. Dari

channel mobile, aplikasi dapat dikembangkan ke banyak *platform* sistem operasi *mobile* dengan *source* yang sama.



Gambar 3.3 Konsep Multi-channel application pada DevExtreme

DevExtreme memanfaatkan beberapa *library* yang tersedia diantaranya JQuery, Knockout, globalize dan PhoneJS. Setiap *reference* terhadap *library* dan *view* ditempatkan pada *file* `index.html`.

DevExtreme juga menyediakan tema yang sesuai dengan platform tujuan, antara lain:

- `dx.android.holo-dark.css`,
- `dx.android.holo-light.css`,
- `dx.ios.default.css`,

- dx.win8.black.css,
- dx.win8.white.css,
- dx.tizen.black.css,
- dx.tizen.white.css,
- dx.generic.light.css, dan
- dx.common.css.

Selain itu, DevExtreme menyediakan layout navigasi yang bisa digunakan diantaranya:

- *Navbar Layout,*
- *SlideOut Layout,*
- *Pivot Layout,*
- *Simple Layout,*
- *Split Layout,* dan
- *Empty Layout.*

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

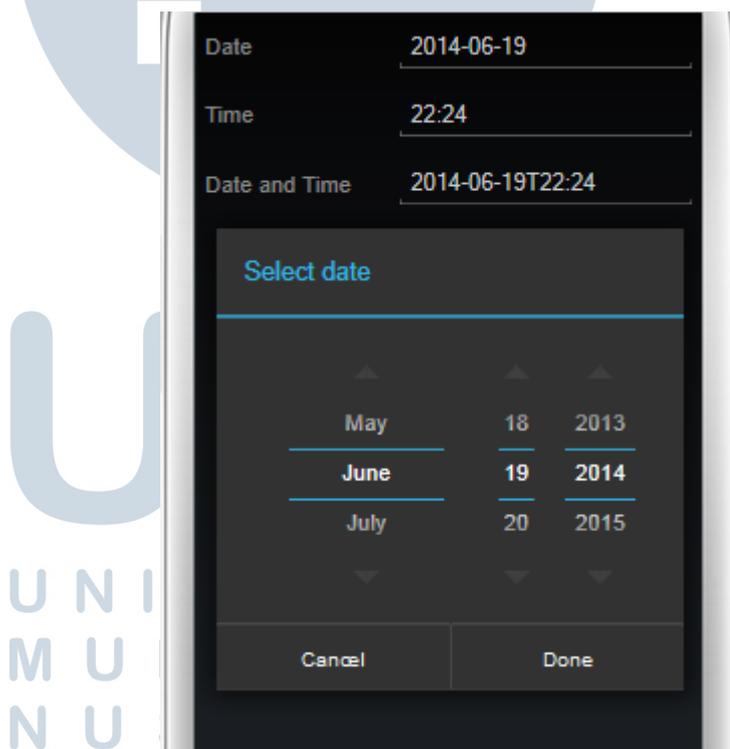


Gambar 3.4 *Navbar Layout* dengan Tema Android *holo-dark*



Gambar 3.5 *SlideOut Layout* dengan Tema Android *holo-dark*

DevExtreme selain menyediakan template navigasi dan tema juga menyediakan sekumpulan UI *widget* yaitu sebuah komponen visual yang ditampilkan browser dari sekumpulan HTML Element, JavaScript code dan CSS stylesheets yang dioptimalisasi untuk penggunaan pada layar sentuh. Salah satu widget yang dioptimasi untuk kebutuhan tertentu adalah dxDateBox pada gambar 3.6.



Gambar 3.6 Contoh dxDateBox

Salah satu fitur yang paling sering digunakan adalah dxList yang dapat dikustomisasi sesuai kebutuhan.

Selain itu kelebihan utama dari penggunaan *tools* DevExtreme adalah aplikasi yang dibangun tidak hanya bisa digunakan di *web* tetapi juga dapat langsung di-*compile* menggunakan PhoneGap menjadi *native package* dari berbagai sistem operasi *mobile* populer seperti android, iOS, dan windows phone yang siap dipublish ke *apps store* masing-masing platform. Kelebihan lainnya adalah setiap aplikasi yang di-*compile* akan memiliki tampilan UI layaknya aplikasi *native* dari sistem operasi *mobile* tersebut.

Beberapa kelebihan lain dari *tools* DevExtreme adalah:

- DevExtreme dapat digunakan dengan diintegrasikan kedalam Microsoft Visual Studio IDE.
- DevExtreme menyediakan *wizard* yang membantu membuat *view* sederhana dari WCF OData service yang digunakan.
- DevExtreme menyediakan *Visual UI editor* yang dapat membantu pengembang termasuk dukungan terhadap *drag-and-drop* dan *design/markup split view*.
- DevExtreme juga menyediakan simulator yang dijalankan dalam *web browser* sehingga pengembang tidak perlu memiliki banyak perangkat.

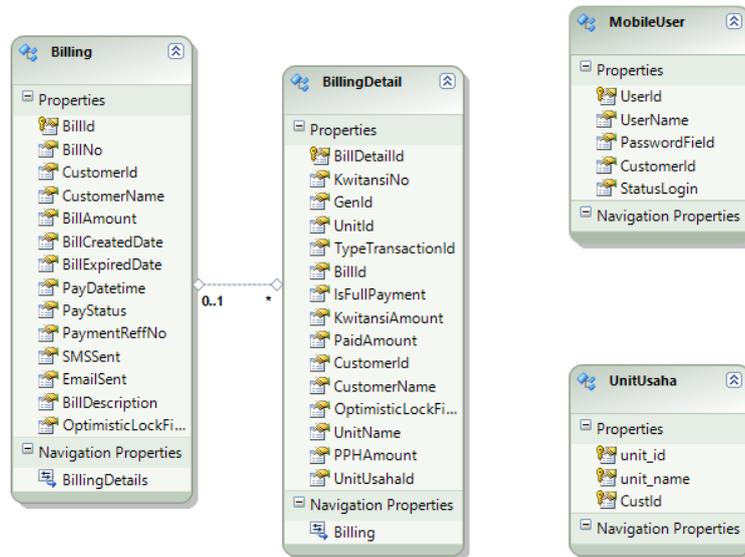
3.3.1.2 Perancangan dan Pengembangan WCF OData Service

WCF OData Service digunakan untuk menghubungkan aplikasi dengan *database*, hal ini dikarenakan aplikasi yang dibangun menggunakan DevExtreme tidak dapat langsung berhubungan dengan *database*.

DevExtreme juga menyediakan template untuk membangun WCF OData Service yang sudah dimodifikasi dengan:

- *Enable JSONP Support*
- *Enable Cross Domain Operation Support*

Untuk memenuhi kebutuhan diperlukan sejumlah tabel untuk menyimpan data-data yang berkaitan dengan transaksi. Pada tabel *Billing* berisi header dari setiap tagihan yang meliputi *BillNo*, *CustomerId*, *BillAmount*, *BillCreatedDate*, *BillExpiredDate*, dll. Sedangkan pada tabel *BillingDetail* digunakan untuk menyimpan data kwitansi yang termasuk ke dalam satu tagihan yang meliputi *KwitansiNo*, *KwitansiAmount*, *PaidAmount*, *CustomerId*, dll. Selain itu dibutuhkan dua entitas tambahan yaitu *MobileUser* yang menyimpan data pengguna aplikasi ini seperti *UserName*, *PasswordField*, *CustomerId*, dll, dan *UnitUsaha* yang merupakan *view* dari sebuah tabel di *database* yang lain. Entity Data Model dari tabel-tabel tersebut dapat dilihat pada gambar 3.6.



Gambar 3.7 Entity Data Model pada WCF OData Service

Selain Entitas yang dipublish pada WCF OData Service terdapat juga entitas yang digunakan dalam sistem ini yang tidak di publish yaitu tabel *BillCounter*.

No	Nama Kolom	Tipe Data	Panjang	Deskripsi
1	AccId	Varchar	10	Menyimpan Nomor Akun untuk setiap unit bisnis
2	TransactionType	Varchar	10	Menyimpan kode tipe transaksi
3	Periode	Varchar	10	Menyimpan data periode per bulan

4	Ctr	Int	-	Nomor terakhir yang tercatat
---	-----	-----	---	------------------------------

Tabel 3.1 Struktur Tabel *BillCounter*

Pada WCF OData Service dibutuhkan Stored Procedure untuk membantu kinerja aplikasi yang perlu di-*publish* untuk bisa digunakan pada aplikasi. Pada WCF OData Service yang dibuat ini terdapat tujuh store procedure yang dipublish yaitu:

1. SP_AuthenticationMobile

Fungsi: Melakukan autentifikasi user yang login dengan mencocokkan data yang dikirim dengan data yang ada di tabel.

Function Import Name: AuthMobile

Parameter: Password dan UserName

Return: CustomerId

2. SP_CreateBillHeader

Fungsi: Membuat row baru dalam tabel Billing dengan beberapa informasi yang sudah tersedia.

Function Import Name: CreateBillHeader

Parameter: CustomerId

Return: BillId

3. SP_LoggingOutMobile

Fungsi: Mengubah status login pada tabel MobileUser sehingga pengguna dapat melakukan login di tempat lain.

Function Import Name: LoggingOut

Parameter: UserName

Return: -

4. SP_TarikBillingDetail

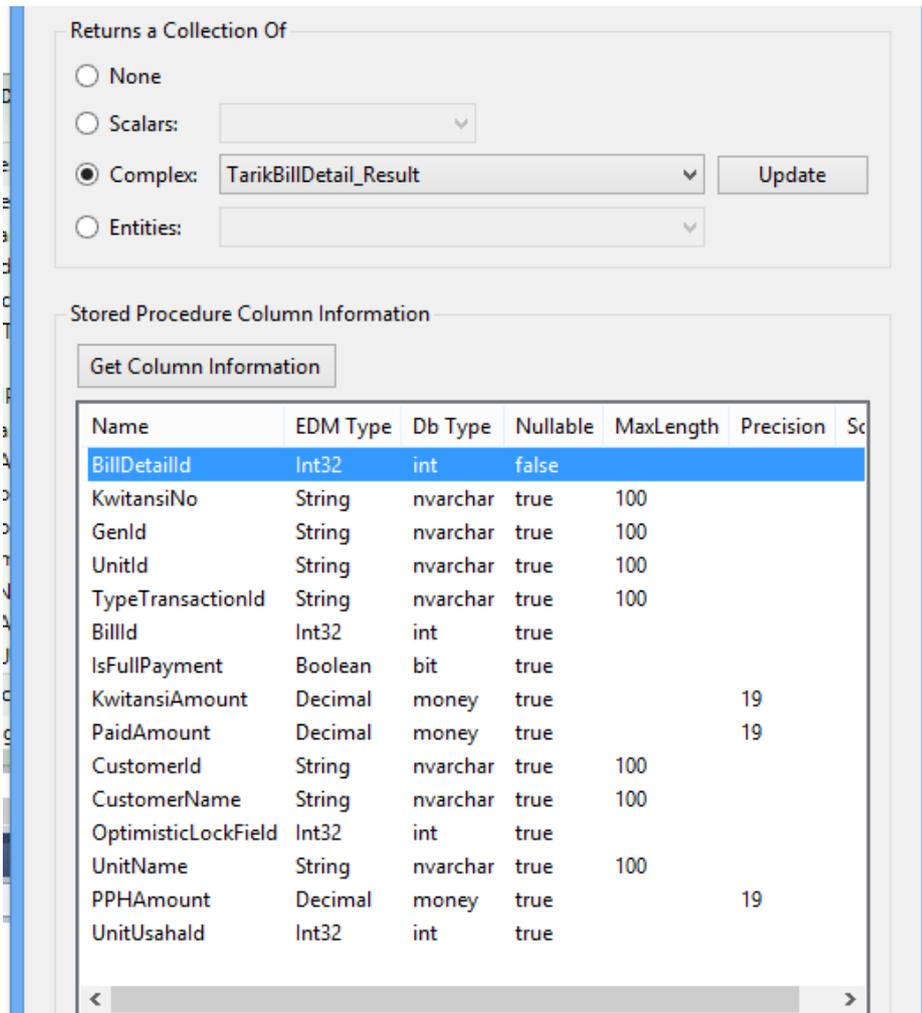
Fungsi: Mengambil koleksi data detail dari sebuah *billing*

Function Import Name: TarikBillDetail

Parameter: BillId

Return: TarikBillDetail_Result





Gambar 3.8 Return Value dari SP_TarikBillDetail

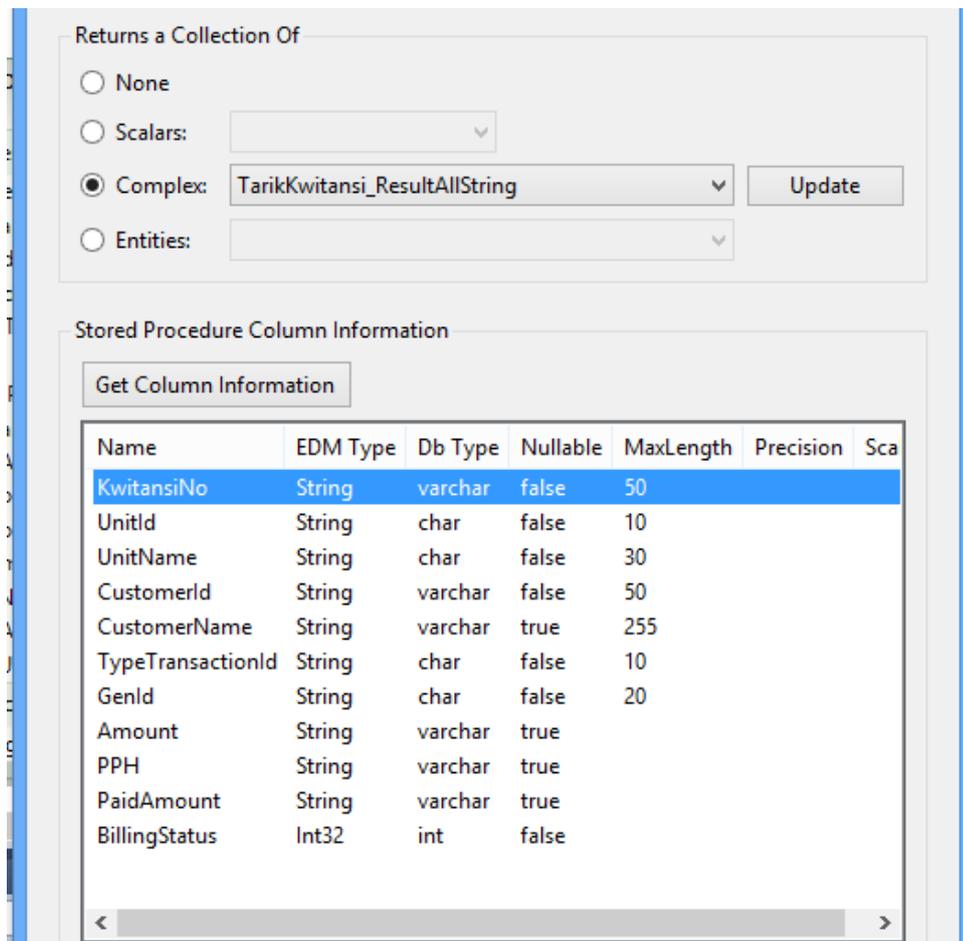
5. SP_TarikKwitansi

Fungsi: Menampilkan koleksi data kwitansi yang belum lunas untuk ditampilkan dalam list kwitansi yang dapat dipilih dalam pembuatan *billing*.

Function Import Name: TarikKwitansi

Parameter: CustomerId

Return: TarikKwitansi_ResultAllString



Gambar 3.9 *Return Value* dari SP_TarikKwitansi

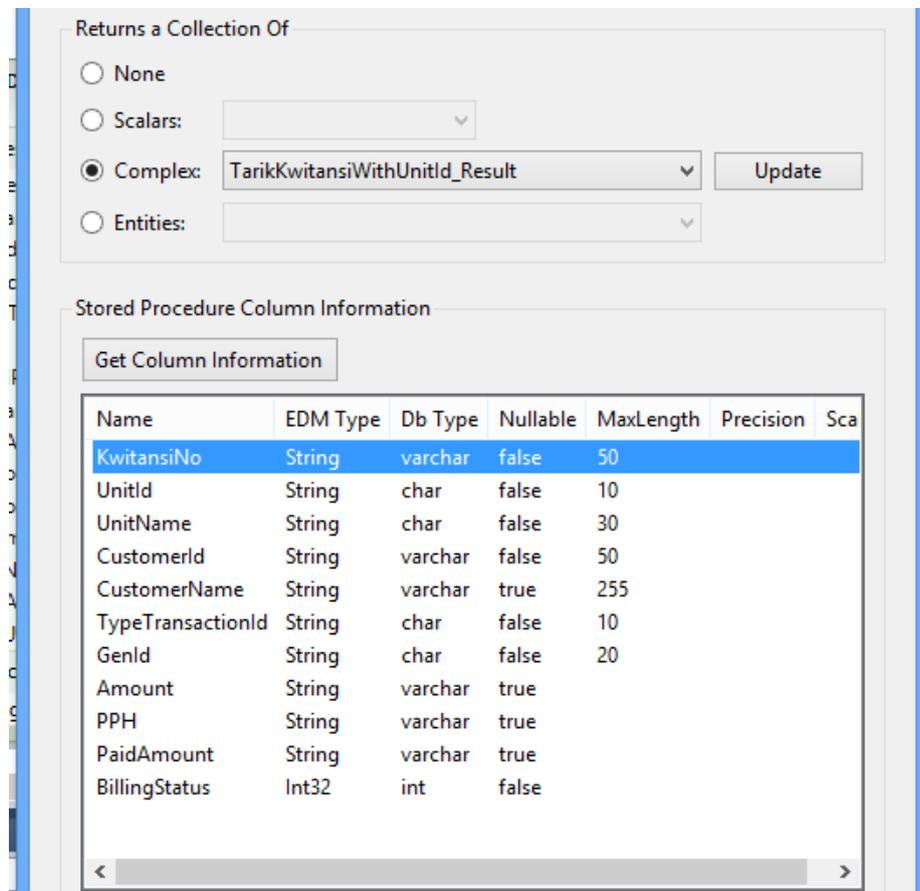
6. SP_TarikKwitansiWithUnitId

Fungsi: Menampilkan koleksi data kwitansi yang belum lunas yang di-*filter* menggunakan UnitId untuk ditampilkan dalam list kwitansi yang dapat dipilih dalam pembuatan *billing*.

Function Import Name: TarikKwitansiWithUnitId

Parameter: CustomerId, UnitId

Return: TarikKwitansiWithUnitId_Result



Gambar 3.10 Return Value dari SP_TarikKwitansiWithUnitId

7. SP_UpdateKwitansiToCMS

Fungsi: Meng-update nilai PendingAmount dan BillingStatus pada database cms yang merupakan schema terpisah dari database sistem ini.

Function Import Name: SP_UpdatingKwitansiToCMS

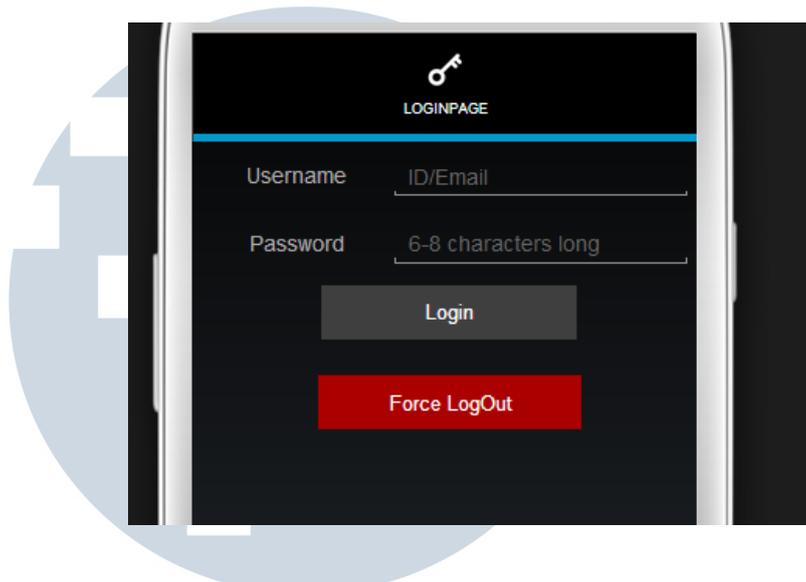
Parameter: CustId, GenId, KwitansiNo, PendingAmount, TypeTransaksi, UnitId

Return: -

3.3.1.3. Pengembangan Aplikasi menggunakan *tools* DevExtreme

Aplikasi yang dibangun terdiri dari beberapa view yaitu:

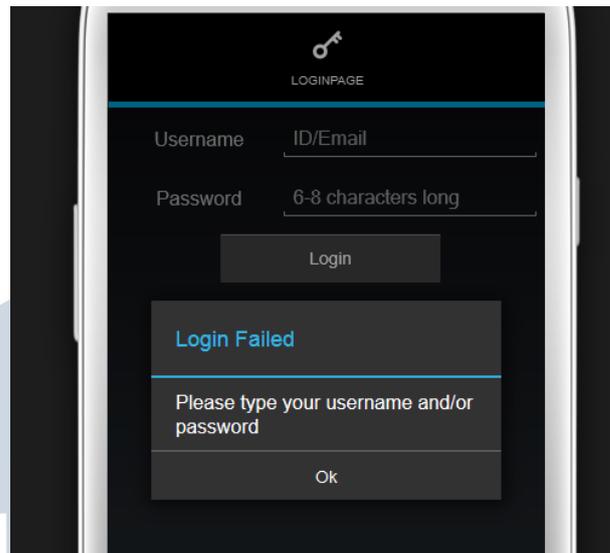
1. Login View



Gambar 3.11 Login View Android

Pada Login View terdapat satu *text field* untuk *Username*, satu *password field*, tombol *Login*, dan tombol *Force Logout*.

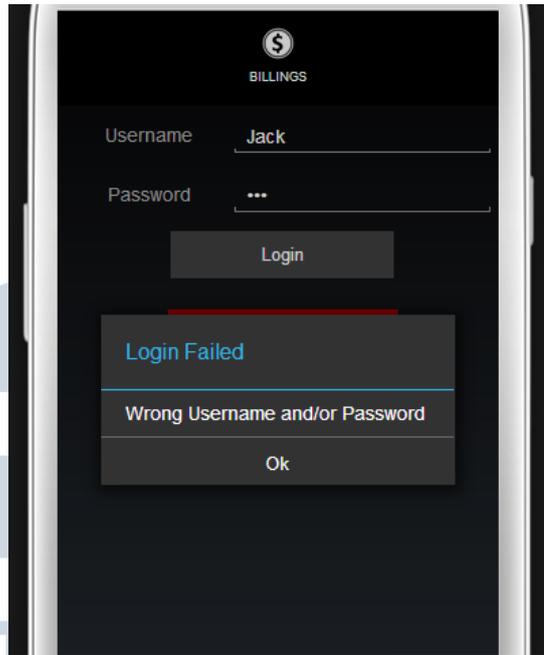
Pengguna harus memasukkan *username* dan *password* ke tempat yang telah disediakan dan menekan salah satu tombol yang ada. Apabila salah satu *field* tidak diisi maka akan muncul *error* seperti Gambar 3.11



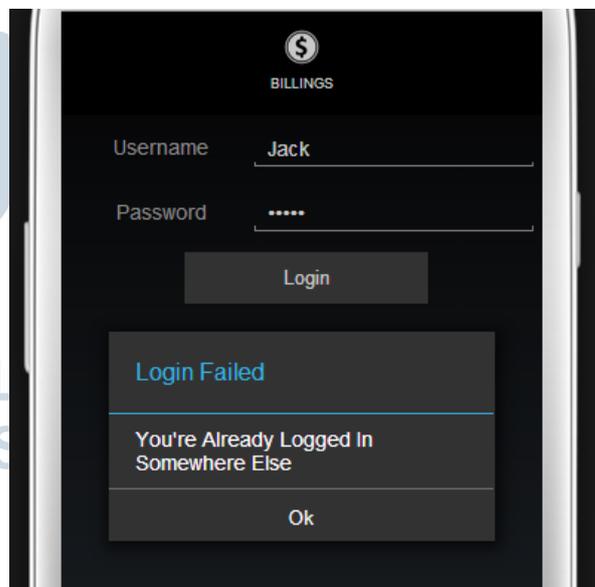
Gambar 3.12 *Error - Field Kosong*

Apabila tombol Login ditekan maka username dan password yang sudah di hash menggunakan SHA2 256bit akan dikirimkan ke WCF OData Service menggunakan metode POST, apabila salah satu dari *username* dan *password* ada yang salah maka akan muncul *error* seperti Gambar 3.12. Apabila *user* tersebut sedang *login* di tempat lain maka akan muncul *error* seperti Gambar 3.13.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

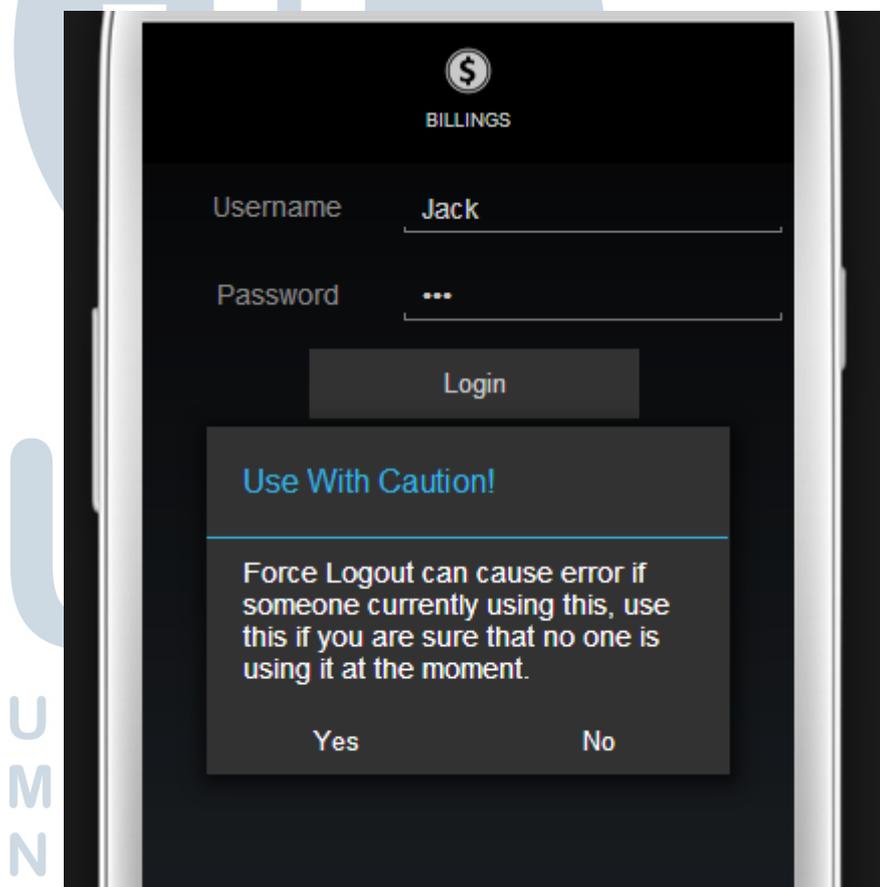


Gambar 3.13 *Error - Salah username atau password*



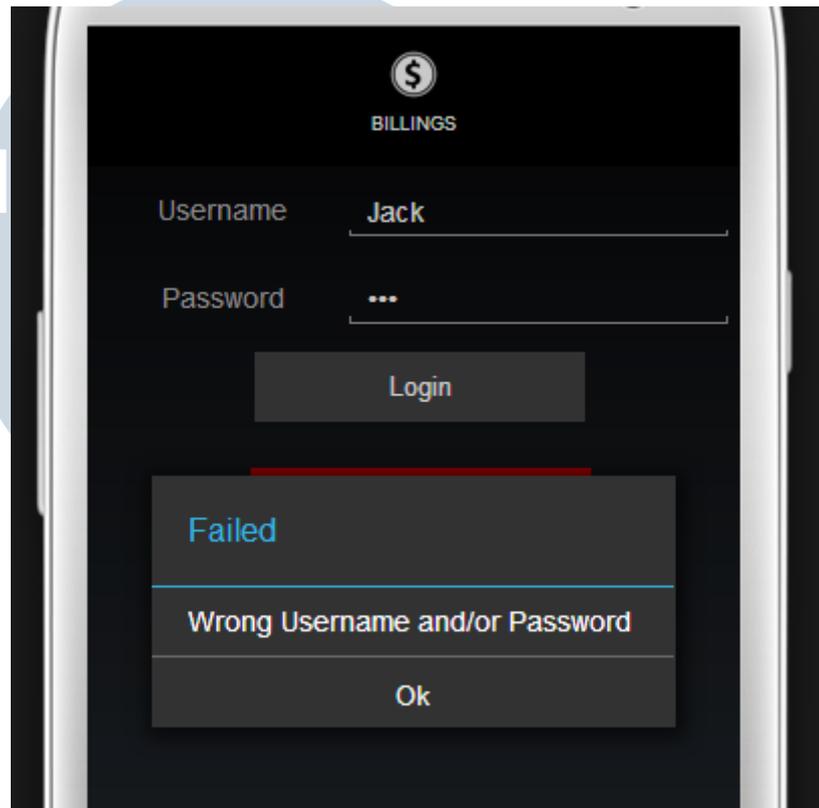
Gambar 3.14 *Error - Sedang ada yang login ditempat lain*

Apabila tombol Force LogOut ditekan, akan muncul *warning* seperti pada Gambar 3.14. Tombol Force LogOut sendiri ditambahkan untuk menghindari kejadian-kejadian yang tidak diinginkan yang menyebabkan status *login* pada *database* tidak *ter-update* dan mengakibatkan *user* tidak dapat melakukan *login*.



Gambar 3.15 Warning saat tombol force logout ditekan

Untuk melakukan Force Logout pengguna juga perlu mengetahui dan mengisi field username serta password. Apabila salah satu field tersebut salah maka akan muncul warning seperti pada Gambar 3.15

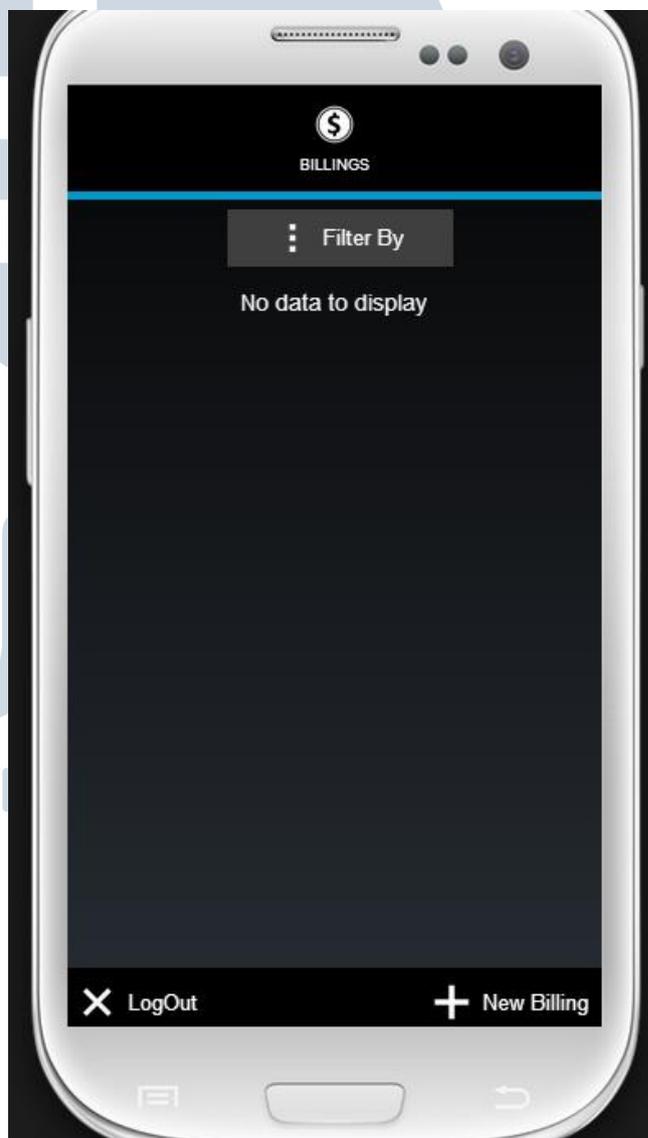


Gambar 3.16 Error - Force Logout Gagal

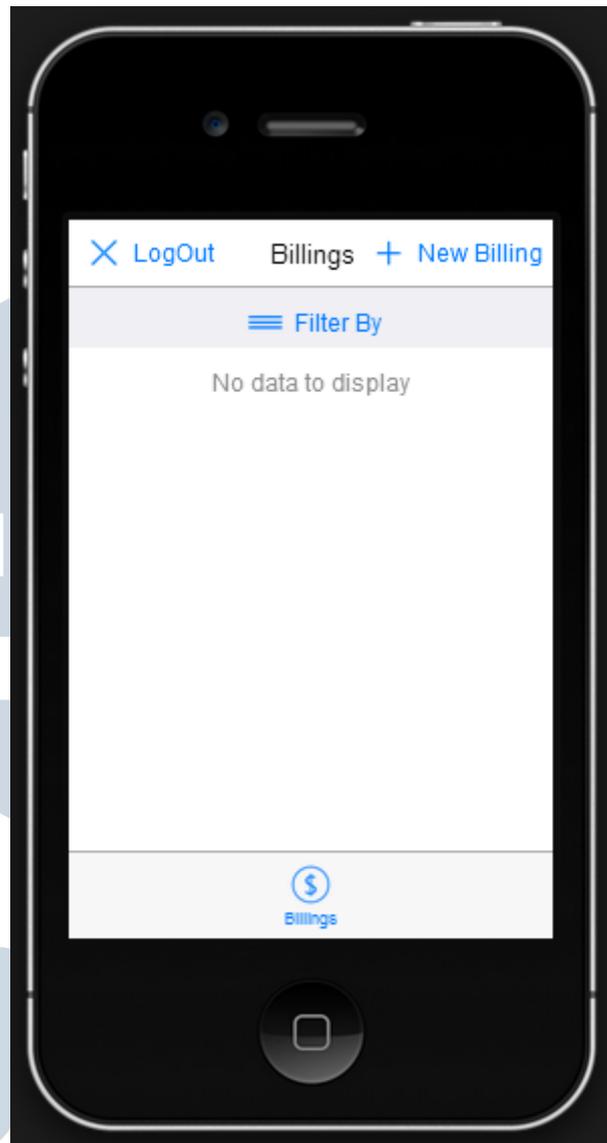
2. Billing View

Setelah sukses melakukan *login*, selanjutnya *billing view* akan ditampilkan. *Billing view* adalah *view* yang mempunyai *list* dari *billing-billing* yang pernah dibuat *user* tersebut dengan kronologi berdasarkan tanggal *billing* dibuat dari yang paling baru ke yang paling lama. Pada *billing view* juga disediakan

filter berupa *drop-down menu* dengan pilihan *All*, *Pending*, *Expired*, dan *Paid*. Selain itu ada tambahan tombol pada billing view yaitu tombol *LogOut* dan tombol *New Billing* pada bagian bawah layar untuk android dan bagian atas layar untuk iOS. Tampilan dari *billing view* dapat dilihat pada gambar-gambar berikut ini:

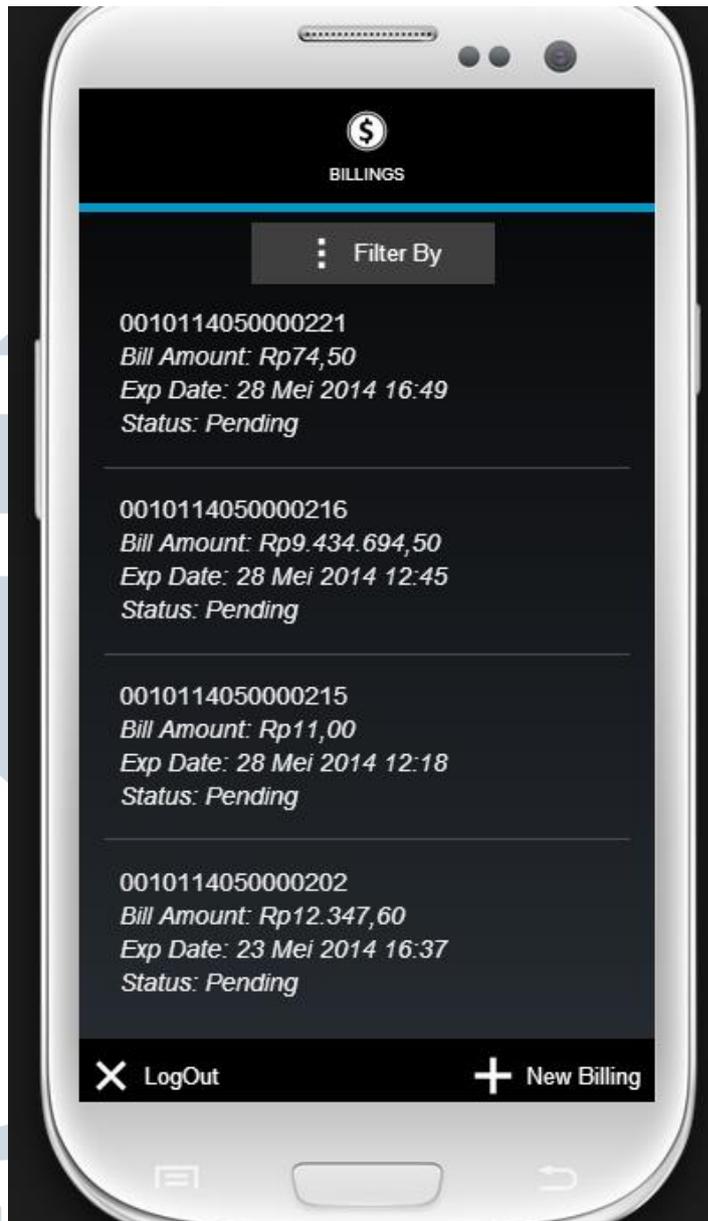


Gambar 3.17 *Billing View* Android - Tanpa Data

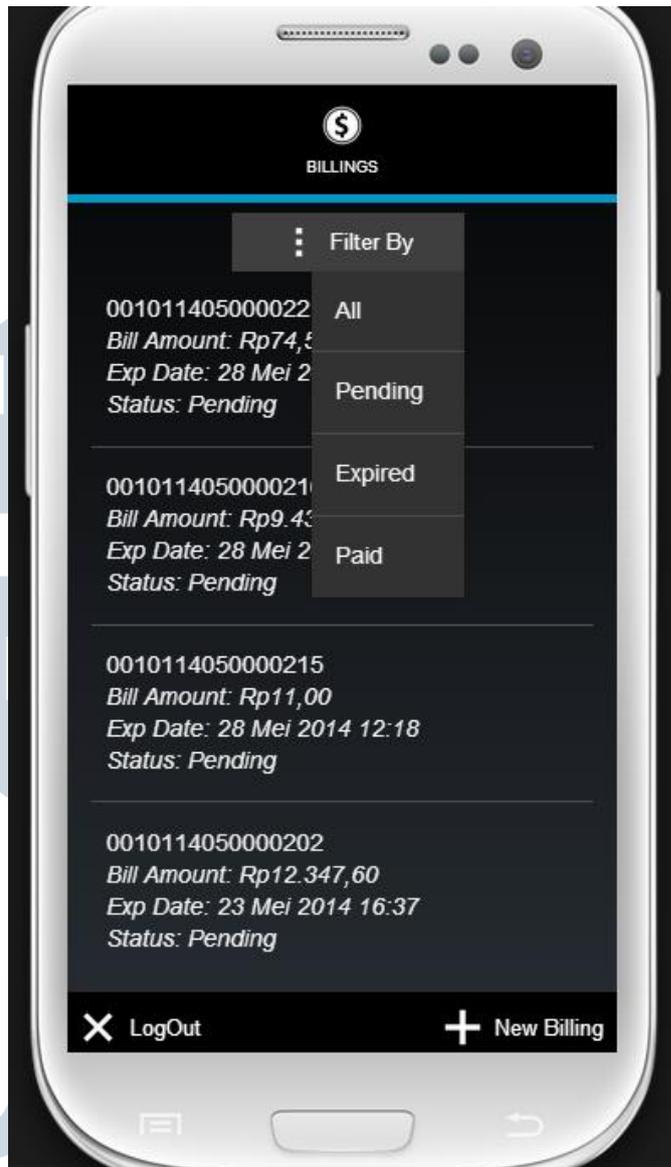


Gambar 3.18 *Billing View* iOS - Tanpa Data

Pada *billing view* setiap objek dalam *list* akan menampilkan data berupa *Bill No*, *Bill Amount*, *Expired Date*, dan *Status*.

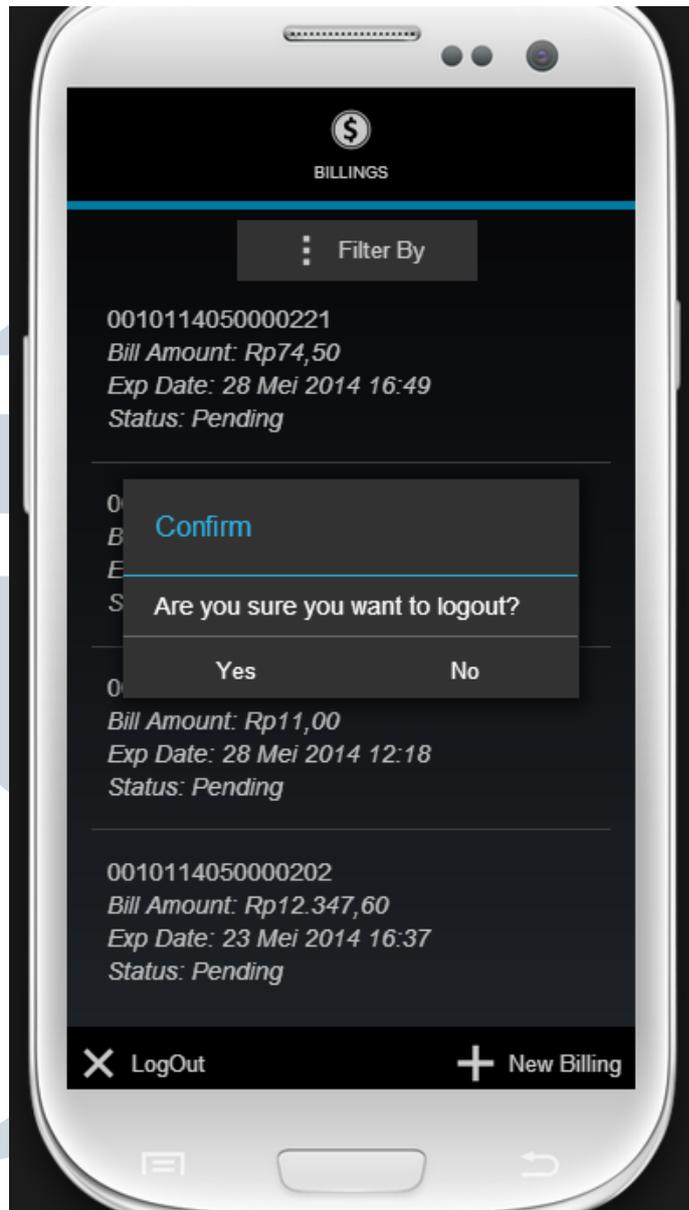


Gambar 3.19 *Billing View* - Dengan Data



Gambar 3.20 *Billing View - Filter menggunakan Drop-down Menu*

Apabila *user* menekan tombol *logout* maka akan muncul tampilan seperti Gambar 3.20

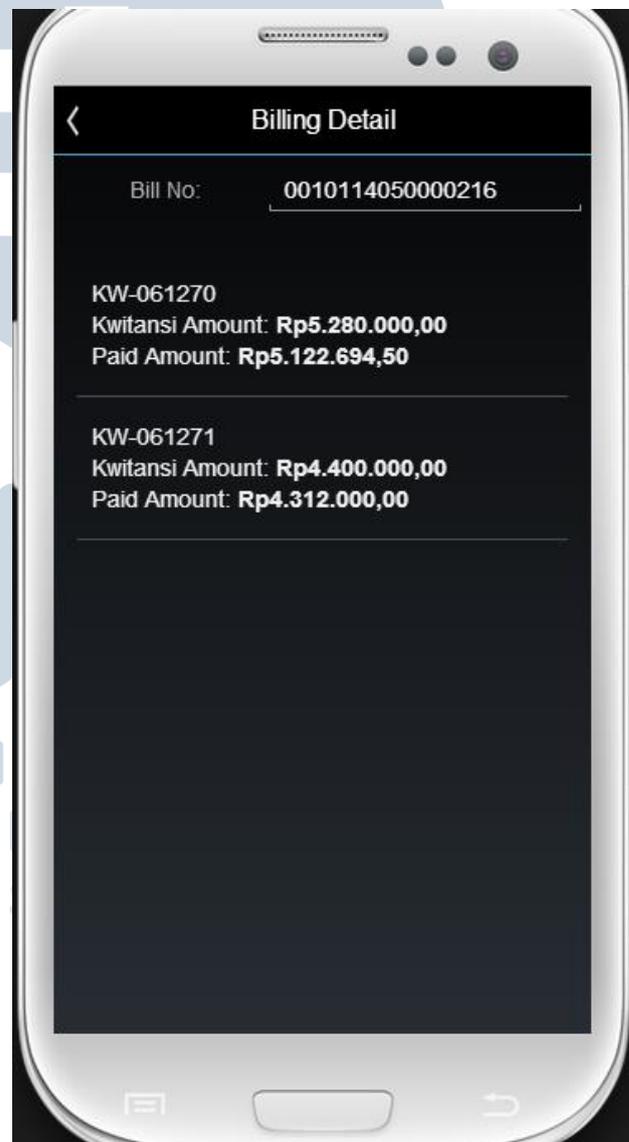


Gambar 3.21 *Billing View - Logout Confirmation Pop-Up*

Setelah Logout, aplikasi akan menavigasi ke login view kembali.

3. Billing Detail View

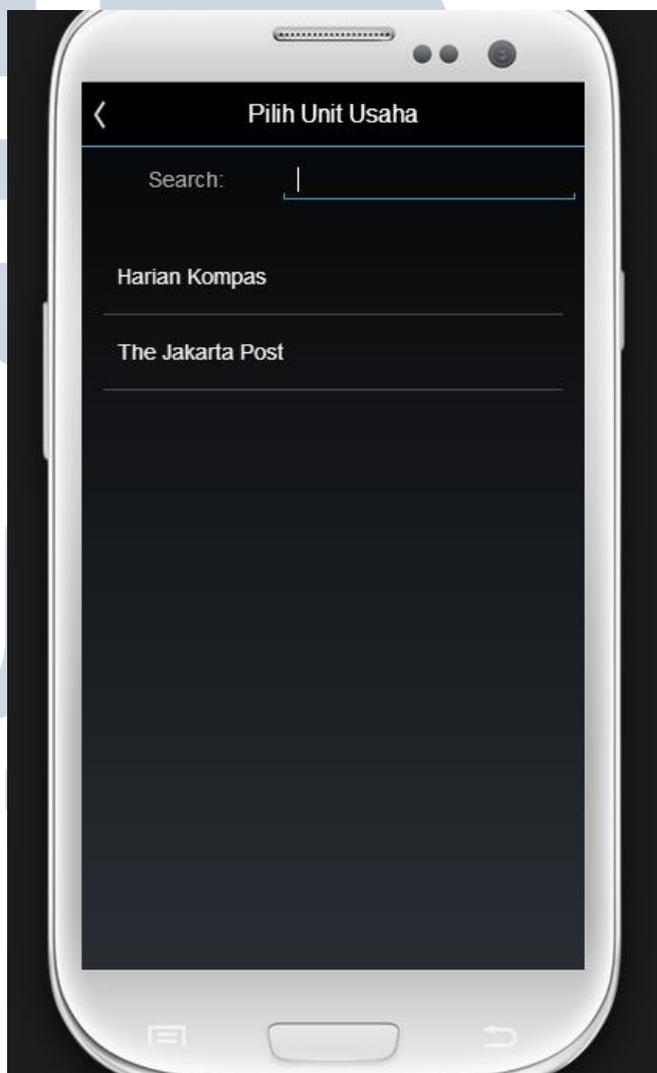
Apabila user memilih menekan objek pada list maka *Billing Detail View* akan muncul yang menampilkan informasi berupa Billing Number, dan list kwitansi yang masuk kedalam billing tersebut. List kwitansi itu menampilkan nomor kwitansi, total nilai kwitansi dan nilai yang dimasukkan kedalam billing tersebut.



Gambar 3.22 *Billing Detail View*

4. Unit Usaha View

Unit Usaha *View* digunakan *user* untuk memilih Unit Usaha dimana *user* memiliki hutang, hal ini dikarenakan setiap unit usaha memiliki rekening masing-masing. Pilihan *user* akan digunakan untuk mem-*filter* kwitansi yang ditampilkan yaitu hanya kwitansi dari unit usaha tersebut.

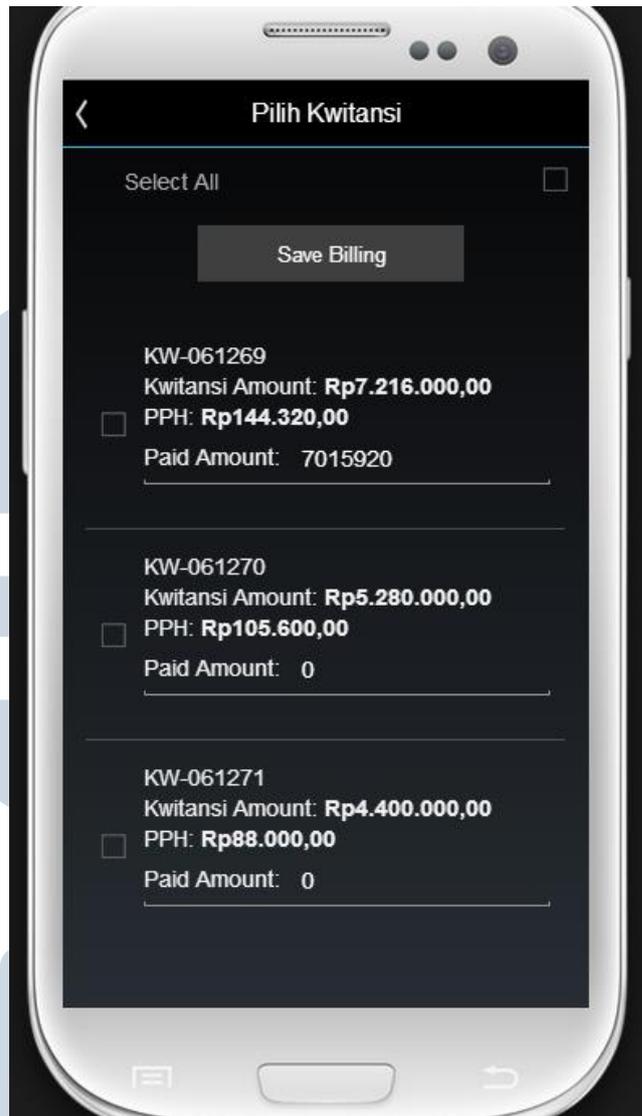


Gambar 3.23 Unit Usaha *View*

5. Create Billing View

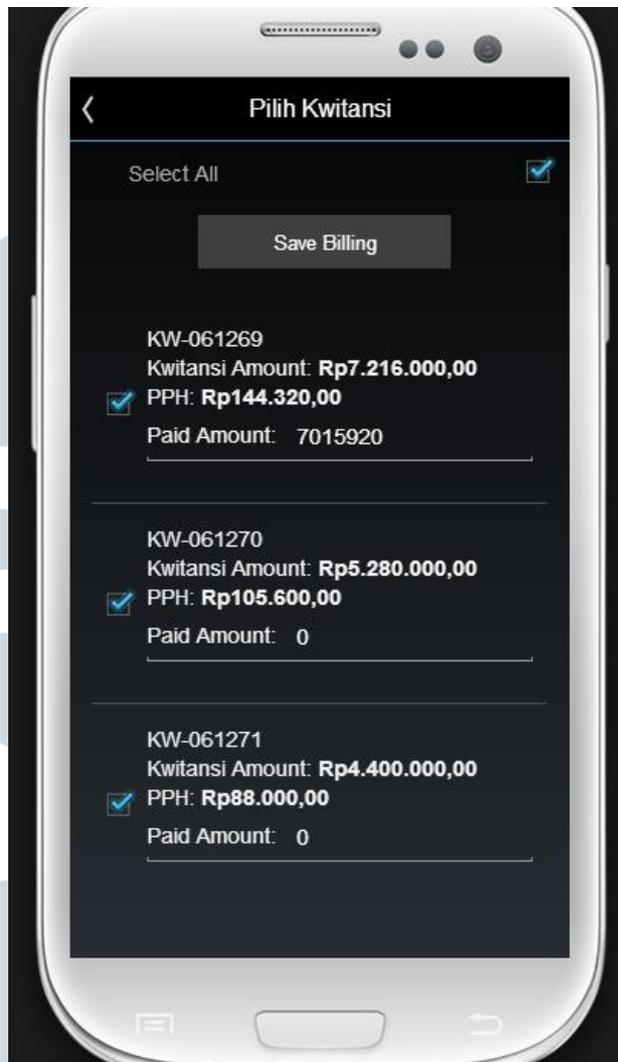
Create Billing View akan menampilkan *list* kwitansi yang masi belum lunas yang sudah di-*filter* sesuai unit usaha yang dipilih di unit usaha *view*. Dalam *Create Billing View* ada pilihan *select all* untuk memilih semua kwitansi yang ada ataupun user bisa memilih satu per satu dengan cara menekan *checkbox* ataupun objek tersebut. Setiap objek memiliki informasi Nomor Kwitansi, Kwitansi *Amount*, PPH, dan *Paid Amount*. Hanya satu *field* yang dapat diisi/edit yaitu *field Paid Amount*, *field* ini adalah *number box* yang hanya bisa diisi dengan angka.





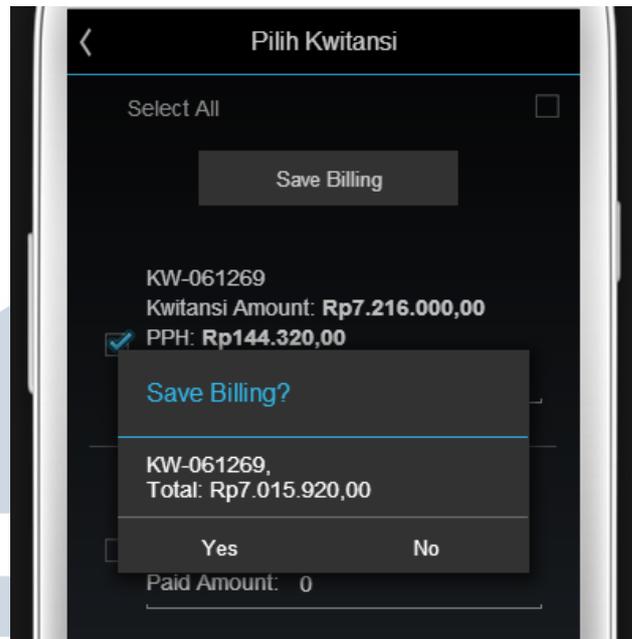
Gambar 3.24 *Create Billing View*

Apabila *checkbox select all* dicentang, maka semua objek kwitansi akan tercentang dan nilai Paid Amountnya akan sesuai nilai maksimal, yaitu nilai kwitansi yang belum dibayar. Hal ini dikarenakan pembayaran kwitansi dapat dilakukan secara sebagian terlebih dahulu

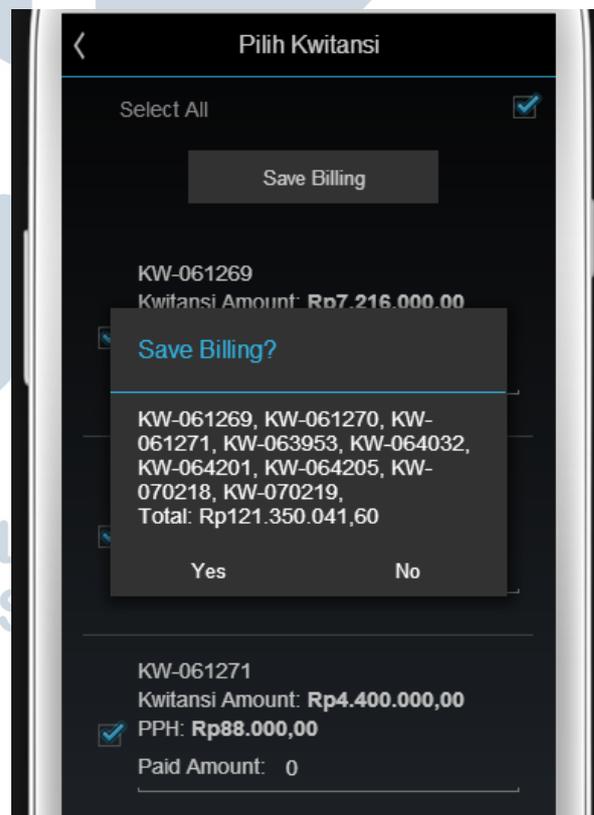


Gambar 3.25 *Create Billing View - Select All*

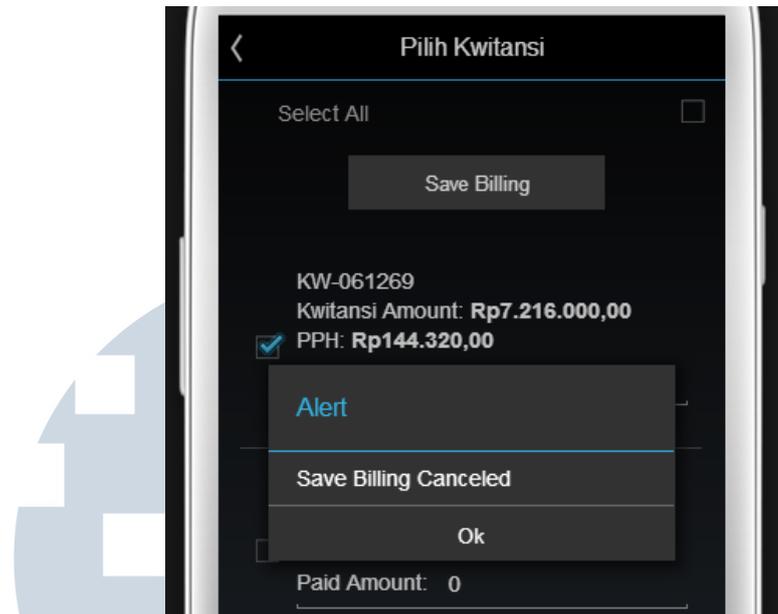
Setelah *user* selesai memilih kwitansi yang akan dibayar *user* harus menekan tombol *Save Billing* untuk menyimpan pilihannya menjadi satu keutuhan *billing*. Saat tombol *Save Billing* ditekan maka akan muncul *pop-up* yang meminta konfirmasi *user*.



Gambar 3.26 *Save Billing Confirmation Pop-up - Satu Kwitansi*



Gambar 3.27 *Save Billing Confirmation Pop-up - Banyak Kwitansi*



Gambar 3.28 Save Billing Cancelled Pop-up

Setelah *Save Billing* berhasil maka *user* akan dinavigasi ke *billing view* kembali.

3.3.1.4. Implementasi dan Pengembangan pada *Development Environment*

Aplikasi dan WCF OData Data Service dibangun dalam development environment dengan spesifikasi:

- Windows 7 32 Bit,
- Microsoft Visual Studio 2010,
- DevExtreme 13.2.9 Untuk Pengembangan aplikasi,
- DevExtreme 13.1.8 Untuk Pengembangan WCF OData Service,
- Database SQL Server 2008,
- Tools SQL Server Management Studio,

- Web Browser Mozilla Firefox dan Google Chrome,
- Perangkat Android Samsung Galaxy Y dengan OS 2.3 Gingerbread, dan
- Perangkat Android Nexus 4 Dengan OS 4.4.2 KitKat.

Pada pengembangannya terdapat beberapa tahap yang terdiri dari:

- Pengembangan WCF OData Service pada localhost diikuti dengan pengembangan aplikasi secara modular,
- Migrasi pengembangan aplikasi dari DevExtreme 13.1.8 ke DevExtreme 13.2.9,
- Penggabungan atau integrasi modul/bagian aplikasi menjadi satu kesatuan aplikasi,
- Mem-publish WCF OData Service pada remote server yang masih ada dalam LAN Development Environment
- Melakukan testing aplikasi baik melalui simulator di web browser maupun di perangkat android.
- Memperbaiki dan menambahkan beberapa fitur yang dibutuhkan.

Pada implementasinya, ditemukan behavior pada perangkat android yang menganggap waktu yang diterima dari WCF OData service sebagai waktu UTC dan mengkonversi waktu tersebut menjadi waktu lokal perangkat itu. Dikarenakan waktu yang disimpan pada database sudah

merupakan waktu lokal, maka *code* yang di-generate secara otomatis pada WCF OData Service dimodifikasi untuk mengurangi waktu dari *database* sebelum dikirim ke aplikasi.

```
Public Property BillExpiredDate() As Nullable(Of Global.System.DateTime)
Get
    Return DateAdd(DateInterval.Hour, -7, CDate(_BillExpiredDate))
End Get
Set
    OnBillExpiredDateChanging(value)
    ReportPropertyChanging("BillExpiredDate")
    _BillExpiredDate = StructuralObject.SetValidValue(value)
    ReportPropertyChanging("BillExpiredDate")
    OnBillExpiredDateChanged()
End Set
End Property
Public Property PayDatetime() As Nullable(Of Global.System.DateTime)
Get
    If _PayDatetime IsNot Nothing Then
        Return DateAdd(DateInterval.Hour, -7, CDate(_PayDatetime))
    Else
        Return _PayDatetime
    End If
End Get
Set
    OnPayDatetimeChanging(value)
    ReportPropertyChanging("PayDatetime")
    _PayDatetime = StructuralObject.SetValidValue(value)
    ReportPropertyChanging("PayDatetime")
    OnPayDatetimeChanged()
End Set
End Property
Public Property BillCreatedDate() As Nullable(Of Global.System.DateTime)
Get
    Return DateAdd(DateInterval.Hour, -7, CDate(_BillCreatedDate))
End Get
Set
    OnBillCreatedDateChanging(value)
    ReportPropertyChanging("BillCreatedDate")
    _BillCreatedDate = StructuralObject.SetValidValue(value)
    ReportPropertyChanging("BillCreatedDate")
    OnBillCreatedDateChanged()
End Set
End Property
```

Gambar 3.29 Mengurangi jam di WCF OData Service

Pada tiga bagian yang ditandai di gambar 3.29, nilai yang di-return untuk ketiga bagian itu dikurangi 7 jam, yang merupakan perbedaan waktu UTC dan waktu lokal server, dari data pada database.

Jadwal pengembangan dapat dilihat pada tabel 3.2 berikut

Jenis Pekerjaan	Minggu								
	1	2	3	4	5	6	7	8	9
Pengenalan Tool dan studi literature HTML5 dan Javascript	█	█							
Pengembangan WCF OData Service			█	█	█	█			
Pengembangan Aplikasi			█	█	█	█	█	█	
Testing dan Review							█	█	█
Dokumentasi								█	█

Tabel 3.2 Jadwal Pengembangan Aplikasi

3.3.2 Kendala yang Ditemukan

Selama periode magang ada beberapa kendala yang ditemui, diantaranya:

- Kurangnya waktu untuk mempelajari dan mengenal *tools* DevExtreme lebih dalam,
- Terjadinya beberapa perubahan yang cukup besar pada tahap awal pengembangan dan munculnya kesalahpahaman dalam *business process* yang menyebabkan penggunaan tabel yang salah,

- Beberapa fitur dalam *tools* DevExtreme yang tidak dapat digunakan pada versi 13.1.8,
- Permasalahan yang terjadi spesifik pada perangkat android yang digunakan seperti ukuran layar yang berbeda-beda yang menyebabkan ada bagian yang tidak tampil pada perangkat dengan layar yang lebih kecil. Selain itu, penggunaan *virtual keyboard* yang berbeda pada perangkat menyebabkan karakter yang seharusnya tidak dapat di-input pada *numberbox* dapat dimasukkan.

3.3.3 Solusi Atas Kendala yang Ditemukan

Solusi yang digunakan untuk mengatasi kendala-kendala yang ditemukan adalah:

- Memanfaatkan solusi sederhana yang terbagi-bagi yang dapat ditemukan pada web support DevExtreme,
- Menjadikan aplikasi web sebagai basis pengembangan aplikasi dengan mengikuti alur kerjanya,
- Melakukan migrasi ke DevExtreme 13.2.9 dalam Virtual Box sehingga pengembangan WCF OData Service tetap dapat dilakukan pada versi 13.1.8