

## APPENDIX



**0.77%** PLAGIARISM  
APPROXIMATELY

### Report #13389059

**INTRODUCTION** Background Lately everyone wants something instant, including in the world of technology. Instant here is in processing or processing large amounts of data. On the front-end, speed is important. It can be fast in loading pages, loading images, processing data, transitions between pages, and so on. Application speed will affect the convenience of the user. The slow speed of an application might be caused by a section for processing large data, it takes time to process it. This large data processing not only cause slow speed an application but also slow down the performance in an application. Slow performance in an application can result in large memory usage, reduced frame rate, lags, large GPU usage, high load times, etc. There are several methods for processing data in React Native. Some of them are FlatList and ScrollView. In this case, testing and analysis will be carried out between FlatList and ScrollView which is the most optimal method for processing large data. Array Map and the Lodash library

REPORT #1338905914 JUL 2021, 11:54 AM

CHECKED  
AUTHOR ANDRE KURNIAWAN

PAGE  
1 OF 32

```

1. import React, { Component } from 'react'
2. import {
3.   Container,
4.   Header,
5.   Item,
6.   Input,
7.   Body,
8.   Left,
9.   Icon,
10. } from 'native-base'
11. import { View, TouchableOpacity, ScrollView, Alert, FlatList,
    ActivityIndicator } from 'react-native'
12. import _ from 'lodash'
13. import { HeaderStack, Text } from 'app/components'
14. import metrics from 'app/config/metrics'
15. import ApiConstants from 'app/api/ApiConstants'
16. import ApiMiddleware from 'app/api/ApiMiddleware'
17. import { ModalPenjualan } from 'app/components'
18. import * as navigationActions from 'app/actions/navigationActions'
19. import NavigationService from 'app/navigation/NavigationService'
20. import styles from './styles'
21.
22. class PencarianView extends Component {
23.   constructor(props) {
24.     super(props);
25.     this.state = {
26.       search      : '',
27.       dataPedagang : [],
28.       offset      : 1,
29.       limit       : 10,
30.       loadingMore  : false,
31.       searchMore   : true
32.     }
33.   }
34.
35.   componentDidMount = () => {
36.     this._getData()
37.   }
38.
39.   _getData = () => {
40.     let mToken = this.props.user.token
41.     let mUrl   = ApiConstants.GetPedagang
42.
43.     ApiMiddleware(mUrl, null, 'GET', mToken)
44.     .then((json) => {
45.       if( json.status == 200 ) {
46.         let mResult = json.res.result == null ? [] :
47.         json.res.result
48.         this.setState({
49.           dataPedagang: [...mResult]

```

```

49.         })
50.     }
51.     else {
52.         alert(JSON.stringify(json))
53.     }
54. })
55. }
56.
57. _getTagihan = (idpedagangkios) => {
58.     let mUrl = ApiConstants.GetTagihanNoreg
59.
60.     let mToken = this.props.user.token
61.
62.     let mData = {
63.         idpedagangkios: idpedagangkios,
64.         idcorporate: this.props.user.idcorporate
65.     }
66.
67.     ApiMiddleware(mUrl, mData, 'POST', mToken)
68.     .then((json) => {
69.         if( json.status == 200 &&
70. json.res.result.listtagihan != null ){
71.             this._submit(json.res.result.listtagihan)
72.             // console.log(mData)
73.         }else{
74.             Alert.alert('Peringatan!', "Tidak ada
75. tagihan/Tagihan sudah dibayar")
76.         }
77.     }).catch((err) => {
78.         Alert.alert(err, json.res)
79.         navigationActions.navigateBack()
80.     })
81. }
82.
83. _submit = (data) => {
84.     let mHeader = {
85.         idcorporate : this.props.user.idcorporate,
86.         tid : this.props.user.tid,
87.         mid : this.props.user.mid,
88.         cid : this.props.user.cid,
89.         namacorporate : this.props.user.namacorporate,
90.         device_id : this.props.user.serialnumber,
91.         iduser : this.props.user.iduser,
92.         merchantkey : this.props.user.merchantkey
93.     }
94.
95.     navigationActions.navigateToDetail({
96.         data : data,
97.         header : mHeader,
98.     })

```

```

99.     _keyExtractor = (item, index) => index.toString()
100.
101.     _loadMore = () => {
102.         this.setState(
103.             prevState => ({
104.                 // offset: prevState.offset + 1,
105.                 // loadingMore: true
106.             }),
107.             () => {
108.                 if (this.state.searchMore == true)
109.                     this._getData()
110.             }
111.         )
112.     }
113.
114.     _renderFooter = () => {
115.         if (!this.state.loadingMore) return null
116.
117.         return (
118.             <View
119.                 style={{
120.                     position: 'relative',
121.                     width: 50,
122.                     height: 50,
123.                     paddingVertical: 20,
124.                     borderTopWidth: 1,
125.                     marginTop: 10,
126.                     marginBottom: 10,
127.                     borderColor: '#000'
128.                 }}
129.             >
130.                 <ActivityIndicator animating size="large" />
131.             </View>
132.         )
133.     }
134.
135.     _checkSession = () => {
136.         let mUrl = ApiConstants.SessionCheck
137.
138.         let mToken = this.props.user.token
139.
140.         ApiMiddleware(mUrl, null, 'GET', mToken)
141.             .then((json) => {
142.                 if ( json.status == 401 ){
143.                     Alert.alert('Peringatan!', 'Sesi anda habis.
144.                         Silahkan masuk kembali')
145.                     this.props.onSetAlertLogout(true)
146.                 }else{
147.                 }
148.             }).catch((err) => {

```

```

149.         Alert.alert('Peringatan!', 'Sesi anda habis.
           Silahkan masuk kembali')
150.         this.props.onSetAlertLogout(true)
151.     })
152. }
153.
154.     _searchFilter = (search) => {
155.         this.setState({
156.             search: search
157.         }, () => {
158.             if (this.state.search.length > 3)
159.                 this._getData()
160.                 this._checkSession()
161.         })
162.     }
163.
164.     render() {
165.         const dataPedagang = this.state.dataPedagang
166.
167.         return (
168.             <Container style={styles.container}>
169.                 <Header style={styles.headerContainer}>
170.                     <Left style={{ flex: 0, paddingHorizontal:
171.                         5, marginRight: 5 }}>
172.                         <Icon                    onPress={() =>
173.                             NavigationService.goBack()} name="md-arrow-back" />
174.                         </Left>
175.                         <Body>
176.                             <Item
177.                                 style={{borderBottomWidth:
178.                                     0}}
179.                                 >
180.                                     <Input
181.                                         onChangeText={ (search)
182.                                             => this._searchFilter(search) }
183.                                         autoFocus={ true }
184.                                         placeholder="Search..."
185.                                         value={ this.state.search
186.                                             }
187.                                     />
188.                                 </Item>
189.                             </Body>
190.                         </Header>
191.
192.                         {/* <ScrollView>
193.                             { dataPedagang.map((o, index) => (

```

```

194.                 borderColor: '#EEEEEE',
195.                 paddingHorizontal: 10,
196.                 paddingVertical: 20
197.             }}
198.             onPress={                ()                =>
this._getTagihan(o.idpedagangkios) }
199.         >
200.         <View style={{ flexDirection:
'row', justifyContent: 'space-between' }}>
201.             <Text text={ o.noreg } />
202.             <Text text={ o.noseri } />
203.         </View>
204.
205.         <View style={{ flexDirection:
'row', justifyContent: 'space-between' }}>
206.             <Text text={ o.namapedagang }
/>
207.             <Text text={ o.namacorporate }
/>
208.         </View>
209.     </TouchableOpacity>
210.     ))
211. }
212. </ScrollView> */}
213.
214. <ScrollView>
215.     { _._map(dataPedagang, (o) =>(
216.     <TouchableOpacity
217.     style={{
218.         width: metrics.screenWidth,
219.         borderTopWidth: 1,
220.         borderColor: '#EEEEEE',
221.         paddingHorizontal: 10,
222.         paddingVertical: 20
223.     }}
224.     onPress={                ()                =>
this._getTagihan(o.idpedagangkios) }
225.     >
226.     <View style={{ flexDirection:
'row', justifyContent: 'space-between' }}>
227.         <Text text={ o.noreg } />
228.         <Text text={ o.noseri } />
229.     </View>
230.
231.     <View style={{ flexDirection:
'row', justifyContent: 'space-between' }}>
232.         <Text text={ o.namapedagang }
/>
233.         <Text text={ o.namacorporate }
/>
234.     </View>
235. </TouchableOpacity>

```

```

236.         ))
237.     }
238. </ScrollView>
239.
240.     {/* <FlatList
241.         data={ dataPedagang ? dataPedagang : [] }
242.         keyExtractor={ this._keyExtractor }
243.         renderItem={ ({ item }) => <ListData
nored={ item.nored } noseri={ item.noseri } namapedagang={
item.namapedagang } namacorporate={ item.namacorporate }
noredFunction={ () => this._getTagihan(item.idpedagangkios) }/>
}
244.         onEndReachedThreshold={ 0.5 }
245.         onEndReached={ this._loadMore }
246.         alwaysBounceVertical={false}
247.         // stickyHeaderIndices={[0]}
248.         // initialNumToRender={ 10 }
249.         // ListFooterComponent={
this._renderFooter }
250.         // onMomentumScrollBegin={ () =>
this.onEndReachedCalledDuringMomentum = false }
251.     /> */}
252.
253.     <ModalPenjualan statusStack={ true }
{...this.props} />
254. </Container>
255. )
256. }
257. }
258.
259. const ListData = (props) => {
260.     return(
261.         <TouchableOpacity
262.             style={{
263.                 width: metrics.screenWidth,
264.                 borderTopWidth: 1,
265.                 borderColor: '#EEEEEE',
266.                 paddingHorizontal: 10,
267.                 paddingVertical: 20
268.             }}
269.             onPress={ props.noredFunction }
270.         >
271.             <View style={{ flexDirection: 'row',
justifyContent: 'space-between' }}>
272.                 <Text text={ props.nored } />
273.                 <Text text={ props.noseri } />
274.             </View>
275.
276.             <View style={{ flexDirection: 'row',
justifyContent: 'space-between' }}>
277.                 <Text text={ props.namapedagang } />
278.                 <Text text={ props.namacorporate } />

```

```

279.         </View>
280.     </TouchableOpacity>
281.     )
282. }
283.
284. PencarianView.propTypes = {
285.
286. }
287. export default PencarianView
288.
289. import React, { Component } from 'react'
290. import {
291.     Container,
292.     Header,
293.     Item,
294.     Input,
295.     Body,
296.     Left,
297.     Icon,
298. } from 'native-base'
299. import { View, TouchableOpacity, ScrollView, Alert, FlatList,
    ActivityIndicator } from 'react-native'
300. import _ from 'lodash'
301. import { HeaderStack, Text } from 'app/components'
302. import metrics from 'app/config/metrics'
303. import ApiConstants from 'app/api/ApiConstants'
304. import ApiMiddleware from 'app/api/ApiMiddleware'
305. import { ModalPenjualan } from 'app/components'
306. import * as navigationActions from
    'app/actions/navigationActions'
307. import NavigationService from
    'app/navigation/NavigationService'
308. import styles from './styles'
309.
310. class PencarianFlatlistView extends Component {
311.     constructor(props) {
312.         super(props);
313.         this.state = {
314.             search          : '',
315.             dataPedagang    : [],
316.             offset          : 1,
317.             limit           : 10,
318.             loadingMore     : false,
319.             searchMore      : true
320.         }
321.     }
322.
323.     componentDidMount = () => {
324.         this._getData()
325.     }
326.
327.     _getData = () => {

```



```

328.         let offset = this.state.offset
329.         let limit = this.state.limit
330.         let search = this.state.search
331.         let mToken = this.props.user.token
332.         let mUrl = ApiConstants.GetPedagang
333.
334.
335.         ApiMiddleware(mUrl, null, 'GET', mToken)
336.         .then((json) => {
337.             console.log(mUrl)
338.             console.log(json)
339.             if( json.status == 200 ) {
340.                 let mResult = json.res.result == null ? [] :
                    json.res.result
341.                 // console.log(mResult)
342.                 this.setState({
343.                     // dataPedagang:
                    [this.state.dataPedagang, ...mResult]
                    dataPedagang: [...mResult]
344.                 })
345.             }
346.         }
347.         else {
348.             if(offset == 1){
349.                 Alert.alert(JSON.stringify(json))
350.             }
351.             else{
352.                 this.setState({
353.                     searchMore: false
354.                 })
355.             }
356.         }
357.     })
358. })
359. }
360.
361. _getTagihan = (idpedagangkios) => {
362.     let mUrl = ApiConstants.GetTagihanNoreg
363.
364.     let mToken = this.props.user.token
365.
366.     let mData = {
367.         idpedagangkios: idpedagangkios,
368.         idcorporate: this.props.user.idcorporate
369.     }
370.
371.     ApiMiddleware(mUrl, mData, 'POST', mToken)
372.     .then((json) => {
373.         if( json.status == 200 &&
            json.res.result.listtagihan != null ) {
374.             this._submit(json.res.result.listtagihan)
375.             // console.log(mData)
376.         }else{

```

```

377.             Alert.alert('Peringatan!', "Tidak ada
tagihan/Tagihan sudah dibayar")
378.             }
379.         }).catch((err) => {
380.             Alert.alert(err, json.res)
381.             navigationActions.navigateBack()
382.         })
383.     }
384.
385.     _submit = (data) => {
386.         let mHeader = {
387.             idcorporate    : this.props.user.idcorporate,
388.             tid             : this.props.user.tid,
389.             mid             : this.props.user.mid,
390.             cid             : this.props.user.cid,
391.             namacorporate   : this.props.user.namacorporate,
392.             device_id       : this.props.user.serialnumber,
393.             iduser          : this.props.user.iduser,
394.             merchantkey     : this.props.user.merchantkey
395.         }
396.
397.         navigationActions.navigateToDetail({
398.             data            : data,
399.             header          : mHeader,
400.         })
401.     }
402.
403.     _keyExtractor = (item, index) => index.toString()
404.
405.     _loadMore = () => {
406.         this.setState(
407.             prevState => ({
408.                 // offset: prevState.offset + 1,
409.                 // loadingMore: true
410.             }),
411.             () => {
412.                 if (this.state.searchMore == true)
413.                     this._getData()
414.             }
415.         )
416.     }
417.
418.     _renderFooter = () => {
419.         if (!this.state.loadingMore) return null
420.
421.         return (
422.             <View
423.                 style={{
424.                     position: 'relative',
425.                     width: 50,
426.                     height: 50,
427.                     paddingVertical: 20,

```

```

428.         borderTopWidth: 1,
429.         marginTop: 10,
430.         marginBottom: 10,
431.         borderColor: '#000'
432.     }}
433. >
434.     <ActivityIndicator animating size="large" />
435. </View>
436. )
437. }
438.
439. _checkSession = () => {
440.     let mUrl = ApiConstants.SessionCheck
441.
442.     let mToken = this.props.user.token
443.
444.     ApiMiddleware(mUrl, null, 'GET', mToken)
445.     .then((json) => {
446.         if( json.status == 401 ){
447.             Alert.alert('Peringatan!', 'Sesi anda habis.
448.             Silahkan masuk kembali')
449.             this.props.onSetAlertLogout(true)
450.         }else{
451.             }
452.         }).catch((err) => {
453.             Alert.alert('Peringatan!', 'Sesi anda habis.
454.             Silahkan masuk kembali')
455.             this.props.onSetAlertLogout(true)
456.         })
457.     }
458. _searchFilter = (search) => {
459.     this.setState({
460.         search: search
461.     }, () => {
462.         if (this.state.search.length > 3)
463.             this._getData()
464.             this._checkSession()
465.         })
466.     }
467.
468.     render() {
469.         const dataPedagang = this.state.dataPedagang
470.
471.         return (
472.             <Container style={styles.container}>
473.                 <Header style={styles.headerContainer}>
474.                     <Left style={{ flex: 0, paddingHorizontal:
475.                     5, marginRight: 5 }}>
476.                         <Icon
477.                             onPress={() =>
478.                                 NavigationService.goBack()} name="md-arrow-back" />

```

```

476.             </Left>
477.             <Body>
478.             <Item
479.                 style={{borderBottomWidth:
480.                     0}}
481.             >
482.                 <Input
483.                     onChangeText={ (search)
484.                     => this._searchFilter(search) }
485.                     autoFocus={ true }
486.                     placeholder="Search..."
487.                     value={ this.state.search
488.                 }
489.                 autoCapitalize="characters"
490.             </Item>
491.             </Body>
492.             </Header>
493.             <FlatList
494.                 data={ dataPedagang ? dataPedagang : [] }
495.                 keyExtractor={ this._keyExtractor }
496.                 renderItem={ ({ item }) => <ListData
497.                     noreg={ item.noreg } noseri={ item.noseri } namapedagang={
498.                     item.namapedagang } namacorporate={ item.namacorporate }
499.                     noregFunction={ () => this._getTagihan(item.idpedagangkios) }/>
500.                 onEndReachedThreshold={ 0.5 }
501.                 onEndReached={ this._loadMore }
502.                 alwaysBounceVertical={false}
503.                 // stickyHeaderIndices={[0]}
504.                 initialNumToRender={ 10 }
505.                 // ListFooterComponent={
506.                 this._renderFooter }
507.                 // onMomentumScrollBegin={ ( ) =>
508.                 this.onEndReachedCalledDuringMomentum = false }
509.             />
510.             <ModalPenjualan statusStack={ true }
511.             {...this.props} />
512.         </Container>
513.     )
514. }
515. }
516. }
517. const ListData = (props) => {
518.     return(
519.         <TouchableOpacity
520.             style={{
521.                 width: metrics.screenWidth,
522.                 borderTopWidth: 1,

```

```

517.         borderColor: '#EEEEEE',
518.         paddingHorizontal: 10,
519.         paddingVertical: 20
520.     }}
521.     onPress={ props.noregFunction }
522. >
523.     <View style={{ flexDirection: 'row',
justifyContent: 'space-between' }}>
524.         <Text text={ props.noreg } />
525.         <Text text={ props.noseri } />
526.     </View>
527.
528.     <View style={{ flexDirection: 'row',
justifyContent: 'space-between' }}>
529.         <Text text={ props.namapedagang } />
530.         <Text text={ props.namacorporate } />
531.     </View>
532. </TouchableOpacity>
533. )
534. }
535.
536. PencarianFlatListView.propTypes = {
537.
538. }
539. export default PencarianFlatListView
540. import React, { Component } from 'react'
541. import {
542.     Container,
543.     Header,
544.     Item,
545.     Input,
546.     Body,
547.     Left,
548.     Icon,
549. } from 'native-base'
550. import { View, TouchableOpacity, ScrollView, Alert, FlatList,
ActivityIndicator } from 'react-native'
551. import _ from 'lodash'
552. import { HeaderStack, Text } from 'app/components'
553. import metrics from 'app/config/metrics'
554. import ApiConstants from 'app/api/ApiConstants'
555. import ApiMiddleware from 'app/api/ApiMiddleware'
556. import { ModalPenjualan } from 'app/components'
557. import * as navigationActions from
'app/actions/navigationActions'
558. import NavigationService from
'app/navigation/NavigationService'
559. import styles from './styles'
560.
561. class PencarianView extends Component {
562.     constructor(props) {
563.         super(props);

```

```

564.         this.state = {
565.             search           : '',
566.             dataPedagang     : [],
567.             offset           : 1,
568.             limit             : 10,
569.             loadingMore      : false,
570.             searchMore       : true
571.         }
572.     }
573.
574.     componentDidMount = () => {
575.         // this._getData()
576.     }
577.
578.     _getData = () => {
579.         let offset = this.state.offset //1
580.         let limit  = this.state.limit //10
581.         let search = this.state.search //'
582.         let mToken = this.props.user.token
583.         let mUrl    = ApiConstants.GetPedagangLimit + '/' +
((offset-1)*limit) + '/' + limit + '/' + 0 + '/' + search
584.
585.         ApiMiddleware(mUrl, null, 'GET', mToken)
586.         .then((json) => {
587.             if( json.status == 200 ) {
588.                 let mResult = json.res.result == null ? [] :
json.res.result
589.                 // console.log(mResult)
590.                 this.setState({
591.                     dataPedagang:
[...this.state.dataPedagang, ...mResult]
592.                     // dataPedagang: [...mResult]
593.                 })
594.             }
595.             else {
596.                 if(offset == 1){
597.                     Alert.alert('Peringatan!',           'Terjadi
kesalahan. mohon diulang kembali')
598.                 }
599.                 else{
600.                     this.setState({
601.                         searchMore: false
602.                     })
603.                 }
604.             }
605.         })
606.     }
607.
608.     _getTagihan = (idpedagangkios) => {
609.         let mUrl = ApiConstants.GetTagihanNoreg
610.
611.         let mToken = this.props.user.token

```

```

612.
613.     let mData = {
614.         idpedagangkios: idpedagangkios,
615.         idcorporate: this.props.user.idcorporate
616.     }
617.
618.     ApiMiddleware(mUrl, mData, 'POST', mToken)
619.     .then((json) => {
620.         if( json.status == 200 &&
        json.res.result.listtagihan != null ){
621.             this._submit(json.res.result.listtagihan)
622.             // console.log(mData)
623.         }else{
624.             Alert.alert('Peringatan!', "Tidak ada
        tagihan/Tagihan sudah dibayar")
625.         }
626.     }).catch((err) => {
627.         Alert.alert(err, json.res)
628.         navigationActions.navigateBack()
629.     })
630. }
631.
632. _submit = (data) => {
633.     let mHeader = {
634.         idcorporate : this.props.user.idcorporate,
635.         tid          : this.props.user.tid,
636.         mid          : this.props.user.mid,
637.         cid          : this.props.user.cid,
638.         namacorporate : this.props.user.namacorporate,
639.         device_id    : this.props.user.serialnumber,
640.         iduser       : this.props.user.iduser,
641.         merchantkey  : this.props.user.merchantkey
642.     }
643.
644.     navigationActions.navigateToDetail({
645.         data      : data,
646.         header    : mHeader,
647.     })
648. }
649.
650. _keyExtractor = (item, index) => index.toString()
651.
652. _loadMore = () => {
653.     this.setState(
654.         prevState => ({
655.             offset: prevState.offset + 1,
656.             // loadingMore: true
657.         }),
658.         () => {
659.             if (this.state.searchMore == true)
660.                 this._getData()
661.         }

```

```

662.     )
663.   }
664.
665.   _renderFooter = () => {
666.     if (!this.state.loadingMore) return null
667.
668.     return (
669.       <View
670.         style={{
671.           position: 'relative',
672.           width: 50,
673.           height: 50,
674.           paddingVertical: 20,
675.           borderTopWidth: 1,
676.           marginTop: 10,
677.           marginBottom: 10,
678.           borderColor: '#000'
679.         }}
680.       >
681.         <ActivityIndicator animating size="large" />
682.       </View>
683.     )
684.   }
685.
686.   _checkSession = () => {
687.     let mUrl = ApiConstants.SessionCheck
688.
689.     let mToken = this.props.user.token
690.
691.     ApiMiddleware(mUrl, null, 'GET', mToken)
692.     .then((json) => {
693.       if ( json.status == 401 ){
694.         Alert.alert('Peringatan!', 'Sesi anda habis.
        Silahkan masuk kembali')
695.         this.props.onSetAlertLogout(true)
696.       }else{
697.
698.       }
699.     }).catch((err) => {
700.       Alert.alert('Peringatan!', 'Sesi anda habis.
        Silahkan masuk kembali')
701.       this.props.onSetAlertLogout(true)
702.     })
703.   }
704.
705.   _searchFilter = (search) => {
706.     this.setState({
707.       search: search
708.     }, () => {
709.       if (this.state.search.length > 3)
710.         this._getData()
711.         this._checkSession()

```



```

712.     })
713.   }
714.
715.   render() {
716.     const dataPedagang = this.state.dataPedagang
717.
718.     return (
719.       <Container style={styles.container}>
720.         <Header style={styles.headerContainer}>
721.           <Left style={{ flex: 0, paddingHorizontal:
722.             5, marginRight: 5 }}>
723.             <Icon                    onPress={()          =>
724.               NavigationService.goBack()} name="md-arrow-back" />
725.           </Left>
726.           <Body>
727.             <Item
728.               style={{borderBottomWidth:
729.                 0}}
730.               >
731.               <Input
732.                 onChangeText={      (search)
733.                 => this._searchFilter(search) }
734.                 autoFocus={ true }
735.                 placeholder="Search..."
736.                 value={ this.state.search
737.                 }
738.                 autoCapitalize="characters"
739.               />
740.             </Item>
741.           </Body>
742.         </Header>
743.         <FlatList
744.           data={ dataPedagang ? dataPedagang : [] }
745.           keyExtractor={ this._keyExtractor }
746.           renderItem={ ({ item }) => <ListData
747.             noreg={ item.noreg } noseri={ item.noseri } namapedagang={
748.             item.namapedagang } namacorporate={ item.namacorporate }
749.             noregFunction={ () => this._getTagihan(item.idpedagangkios) }/>
750.           }
751.           onEndReachedThreshold={ 0.5 }
752.           onEndReached={ this._loadMore }
753.           alwaysBounceVertical={false}
754.           // stickyHeaderIndices={[0]}
755.           // initialNumToRender={ 10 }
756.           //
757.           ListFooterComponent={
758.             this._renderFooter }
759.           // onMomentumScrollBegin={      ()          =>
760.           this.onEndReachedCalledDuringMomentum = false }
761.         />

```

```

752.         <ModalPenjualan    statusStack={    true    }
      {...this.props} />
753.     </Container>
754.     )
755. }
756. }
757.
758. const ListData = (props) => {
759.     return(
760.         <TouchableOpacity
761.             style={{
762.                 width: metrics.screenWidth,
763.                 borderTopWidth: 1,
764.                 borderColor: '#EEEEEE',
765.                 paddingHorizontal: 10,
766.                 paddingVertical: 20
767.             }}
768.             onPress={ props.noregFunction }
769.         >
770.             <View    style={{    flexDirection:    'row',
      justifyContent: 'space-between' }}>
771.                 <Text text={ props.noreg } />
772.                 <Text text={ props.noseri } />
773.             </View>
774.
775.             <View    style={{    flexDirection:    'row',
      justifyContent: 'space-between' }}>
776.                 <Text text={ props.namapedagang } />
777.                 <Text text={ props.namacorporate } />
778.             </View>
779.         </TouchableOpacity>
780.     )
781. }
782.
783. PencarianView.propTypes = {
784.
785. }
786.
787. export default PencarianView
788.
789.

```