

SENCHA UNTUK PEMULA



Dr. Arbi Haza Nasution, M.IT
Winda Monika, M.LIS
Salhazan Nasution, M.IT

Pemrograman Mobile HTML5 SENCHA UNTUK PEMULA

Dr. Arbi Haza Nasution, M.IT

Winda Monika, M.LIS

Salhazan Nasution, M.IT

Penerbit:



UNIVERSITAS ISLAM RIAU

Pemrograman Mobile HTML5

SENCHA UNTUK PEMULA

Penulis:

Dr. Arbi Haza Nasution, M.IT
Winda Monika, M.LIS
Salhazan Nasution, M.IT

ISBN: 978-979-3793-86-3

Editor: Dr. Arbi Haza Nasution, M.IT

Penyunting: Winda Monika, M.LIS

Desain sampul dan Tata Letak: Salhazan Nasution, M.IT

Penerbit: Universitas Islam Riau (UIR) Press

Redaksi:

Gedung Serba Guna Universitas Islam Riau
Jl. Kaharuddin Nasution No. 113
Marpoyan 28284, Pekanbaru, Riau, Indonesia
Telp: (0761) 674674
Fax: (0761) 674834
Email: uirpress@uir.ac.id

Cetakan pertama, Oktober 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan
dengan cara apapun tanpa ijin tertulis dari penerbit



KATA PENGANTAR

Puji syukur penulis haturkan kehadirat Allah SWT yang telah memberikan rahmat dan karunianya sehingga penulis bisa menyelesaikan buku ini tanpa ada halangan yang berarti. Buku ini dilatarbelakangi kecintaan penulis dalam mengajar dan mendedikasikan ilmu di bidang teknologi informasi dimana penulis merupakan dosen di jurusan Teknik Informatika, Universitas Islam Riau dan Teknik Informatika Universitas Riau.

Aplikasi Web dan Mobile saat ini menjadi teknologi penting yang digunakan diberbagai macam sektor kehidupan seperti bisnis, pendidikan, olah-raga dan lain sebagainya. Kebutuhan akan teknologi ini menuntut mahasiswa, developer, dan berbagai kalangan untuk mampu menguasainya.

Sencha merupakan sebuah *web framework Javascript* yang mendukung pembangunan aplikasi web atau aplikasi mobile berbasis HTML5. *Framework* ini cukup lengkap hanya saja minim dokumentasi, untuk itu buku ini diharapkan dapat memenuhi dahaga para pembelajar terutama para pemula. Buku ini dibagi ke dalam berbagai



bagian diantaranya pengenalan konsep Sencha cmd, instalasi Sencha cmd, pengenalan Sencha library dan konektivitas ke database.

Dalam penulisan buku ini penulis ucapkan terimakasih kepada berbagai pihak yang telah membantu baik secara materil maupun imateril. Tak lupa pula penulis persembahkan dan dedikasikan buku ini untuk anak kami tercinta Akira Zafirarda Nasution.

Penulis berharap buku ini dapat bermanfaat bagi banyak developer pemula untuk menciptakan kreasi teknologi baru sehingga dapat membantu kelancaran dan kelangsungan hidup orang banyak. Penulis menyadari bahwa buku ini masih jauh dari kesempurnaan oleh karena itu, masukan konstruktif sangat penulis harapkan guna penyempurnaan buku ini.

Pekanbaru, 2019

Tim Penulis



DAFTAR ISI

| | |
|---|------------|
| KATA PENGANTAR | <i>iv</i> |
| DAFTAR ISI | <i>vi</i> |
| DAFTAR GAMBAR | <i>vii</i> |
| 1. PENGENALAN SENCHA | <i>1</i> |
| Prasyarat | <i>5</i> |
| Instalasi..... | <i>5</i> |
| Workstation..... | <i>17</i> |
| Membuat Kerangka Aplikasi | <i>18</i> |
| Menjalankan Aplikasi | <i>22</i> |
| Latihan..... | <i>24</i> |
| 2. MODEL, VIEW, CONTROLLER | <i>25</i> |
| Pengenalan MVC..... | <i>25</i> |
| Membuka Kode Aplikasi dengan Sublime Text | <i>27</i> |
| index.html..... | <i>28</i> |
| Application.js | <i>29</i> |
| app.js..... | <i>30</i> |
| View..... | <i>31</i> |
| View Controller (MainController)..... | <i>35</i> |
| View Model (MainModel)..... | <i>37</i> |
| Store | <i>37</i> |
| Latihan..... | <i>40</i> |
| 3. KOMPONEN ANTARMUKA PENGGUNA | <i>41</i> |
| Layout and Container | <i>41</i> |
| Carousel..... | <i>44</i> |
| Grids | <i>47</i> |
| List..... | <i>48</i> |
| Forms | <i>49</i> |
| Latihan..... | <i>56</i> |
| 4. PERTUKARAN DATA DENGAN JSON | <i>57</i> |
| Pengenalan JSON | <i>57</i> |
| CRUD..... | <i>62</i> |
| Latihan..... | <i>82</i> |
| GLOSARIUM | <i>83</i> |



| | |
|------------------------------|-----------|
| INDEKS..... | 88 |
| BIOGRAFI PENULIS..... | 90 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1. Java Version..... | 5 |
| Gambar 2. Sencha Cmd Type Windows | 6 |
| Gambar 3. Sencha Cmd Setup Wizard | 6 |
| Gambar 4. Sencha Cmd Setup License Agreement..... | 7 |
| Gambar 5. Sencha Cmd Setup Select Directory..... | 7 |
| Gambar 6. Sencha Cmd Select Components | 8 |
| Gambar 7. Select Additional Tasks | 8 |
| Gambar 8. Installing | 9 |
| Gambar 9. Completing the sencha cmd setup wizard | 9 |
| Gambar 10. Browse Windows..... | 10 |
| Gambar 11. System Properties | 11 |
| Gambar 12. Environment variables | 12 |
| Gambar 13. Edit Environment Variables | 13 |
| Gambar 14. Sencha Cmd Version | 14 |
| Gambar 15. Sub folder Ext Js..... | 15 |
| Gambar 16. Sencha Web Start..... | 16 |
| Gambar 17. Sencha Ext JS 6.2.0 Example | 16 |
| Gambar 18. Ext JS Kitchen Sink..... | 17 |
| Gambar 19. Pembuatan Folder Sencha Workstation..... | 18 |
| Gambar 20. Pembuatan Kerangka Aplikasi Ext Js..... | 18 |
| Gambar 21. Kerangka Aplikasi "MyApp"..... | 20 |
| Gambar 22. Struktur Direktori "MyApp"..... | 21 |
| Gambar 23. Aplikasi dan Interface Sencha Cmd | 23 |
| Gambar 24. MVC : Model View Controller..... | 25 |
| Gambar 25. Folder "MyApp"..... | 28 |
| Gambar 26. Index.html..... | 29 |
| Gambar 27. Pesan Konfirmasi..... | 36 |
| Gambar 28. Panel | 44 |
| Gambar 29. Carousel | 45 |



| | |
|---|----|
| Gambar 30. Carousel “Login” | 47 |
| Gambar 31. List Grid..... | 48 |
| Gambar 32. Carousel “List” | 48 |
| Gambar 33. Forms | 56 |
| Gambar 34. View Respons | 62 |
| Gambar 35. Tampilan Aplikasi | 78 |
| Gambar 36. Tambah Data..... | 79 |
| Gambar 37. Pesan Konfirmasi | 80 |
| Gambar 38. Update Data | 80 |
| Gambar 39. Data Setelah di Update | 81 |
| Gambar 40. Delete Data | 81 |
| Gambar 41. Data Setelah di hapus..... | 82 |



1. PENGENALAN SENCHA

Sencha merupakan sebuah web framework Javascript yang mendukung pembangunan aplikasi web atau aplikasi mobile berbasis HTML5. Framework ini dapat digunakan oleh *developer* untuk membangun antarmuka pengguna untuk aplikasi mobile web seperti layaknya aplikasi *native* pada perangkat seluler yang mendukung.

Sencha didasari pada standardisasi perangkat pembuatan web seperti HTML5, CSS3 dan JavaScript. Framework ini bertujuan untuk memfasilitasi pengembangan aplikasi mobile berbasis HTML5 yang dijalankan pada perangkat Android, iOS, Windows, Blackberry, dsb.

Sencha Touch adalah sebuah produk dari Sencha yang dibuat dari penggabungan Ext JS, jQTouch and Raphaël. Sencha Touch pertamakali dirilis pada 17 Juli 2010 yaitu versi 0.90 beta. Versi ini mendukung dijalankan pada berbagai perangkat Android dan iOS (on iPhone, iPod touch, iPad). Selanjutnya, versi stabil yaitu 1.0 dirilis pada November 2010. Versi 1.1.0 kemudian mendukung untuk dijalankan pada aplikasi BlackBerry OS 6.0. Sencha Touch



terbaru yaitu Sencha Touch 2.4.2 dirilis pada Juni 2015 dan dapat dijalankan pada banyak browsers dan platforms seperti Android browser, Google Chrome untuk Android, BlackBerry 10, Bada Mobile Browser, Kindle Fire Browser, Tizen, Windows Phone 8, Windows 8 IE10, dan Mobile Safari.

Sencha Cmd adalah seperangkat *command line* lintas platform atau dikenal dengan *library* yang menyediakan fitur lengkap mencakup *lifecycle management* mulai dari pembuatan proyek baru sampai dengan menjalankan (*deploy*) aplikasi yang kemudian dikemas menjadi suatu produk. *Lifecycle management* mencakup rancangan (*scaffolding*), minifikasi koding, pembuatan aplikasi dan lain sebagainya.

Sencha Cmd merupakan alat dengan fitur lengkap sehingga dapat menghemat waktu dalam membangun sebuah aplikasi mobile. Sencha Cmd terintegrasi dengan Sencha Ext JS dan kerangka kerja Sencha Touch. Sencha Cmd memiliki kemampuan diantaranya.

- **Menyediakan kode:** koding yang tersedia di dalam library Sencha memudahkan *developer* untuk membuat aplikasi mobile baru dan koding tersebut juga dapat



dimodifikasi dengan komponen Model, View dan Controller yang dimiliki oleh Sencha Cmd.

- **JS Compiler:** memiliki JavaScript compiler yang dapat memahami kompleksitas semantik dari kerangka kerja Sencha dan mampu menghasilkan aplikasi dengan ukuran kecil sehingga mengurangi waktu memuat (*loading*) aplikasi.
- **Web Server:** Menyediakan web server ringan yang dapat diakses dari localhost.
- **Paket Management Sistem:** Paket management system didistribusikan untuk memudahkan integrasi seperti Ext JS Themes
- **Sencha Web Application Manager Integration:** dengan mudah mempublikasikan aplikasi baru untuk Sencha Web Application Manager server.
- **Management Tempat Kerja:** membantu dalam berbagi kerangka kerja, paket dan pembuatan koding lintas banyak aplikasi.
- **Membangun Skrip Kode:** menyediakan koding yang dapat digenerasi untuk pembuatan aplikasi, selain itu juga developer dapat mengembangkan lebih lanjut dengan memodifikasi koding/menyesuaikannya dengan



kebutuhan aplikasi.

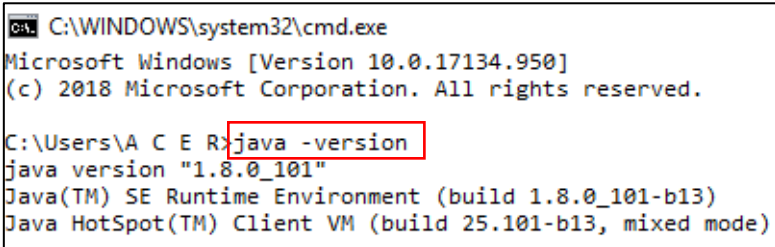
- **Integrasi Cordova / PhoneGap:** Paket native untuk mengalih bentukkan web aplikasi menjadi aplikasi mobile yang sesuai dengan fungsi device/alat dan dapat didistribusikan di dalam App Stores.
- **Menangkap gambar:** Mengonversi fitur CSS3 (seperti radius batas dan gradien linier) menjadi sprite (computer grafis dengan layer bergerak) untuk browser lawas.
- **Tuning Tools:** Alat pemilihan kode yang kuat untuk menyetel apa yang termasuk dalam build akhir aplikasi Anda, menentukan kode umum di seluruh halaman dan partisi kode bersama menggunakan operasi set level tinggi untuk mendapatkan build persis seperti yang Anda inginkan.
- **Konfigurasi Sistem yang Flexible:** mengizinkan defaults untuk menspesifikasikan opsi perintah pada aplikasi atau level tempat kerja atau terhadap semua tempat kerja pada mesin.
- **Logging:** Pencatatan yang kuat untuk membantu Anda memahami cara kerja perintah dan memfasilitasi pemecahan masalah.



- **Perangkat Lunak pihak ketiga:** Untuk Sencha Touch dan Ext JS 5 dan versi yang lama, Sencha Cmd menyertakan versi Kompas dan Sass yang kompatibel.
- **Code Generation Hooks:** Dapat dikhususkan untuk satu halaman atau dibagikan oleh semua halaman di ruang kerja, misalnya, untuk memeriksa konvensi dan pedoman pengkodean saat model baru dibuat).

Prasyarat

Sebelum mulai menggunakan Sencha, terlebih dahulu pastikan bahwa pc Anda sudah terinstalasi java dengan mengetikkan “java -version” pada terminal (Windows) atau *command prompt* (Windows).



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\A C E R>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) Client VM (build 25.101-b13, mixed mode)
```

Gambar 1. Java Version

Instalasi

Beberapa tahap yang dilakukan saat instalasi antara lain.

1. Download Sencha Cmd 6.6.0.13 dari tautan berikut dan



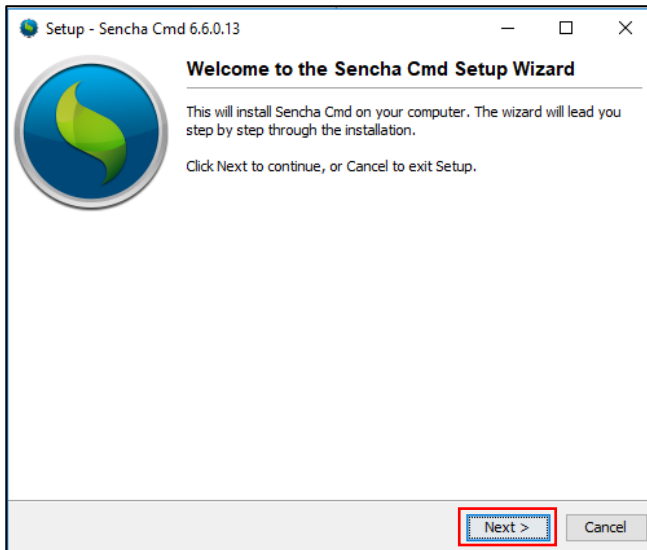
sesuaikan dengan system type windows 32-bit atau 64-bit pada masing-masing windows :

<https://www.sencha.com/products/extjs/cmd-download/>



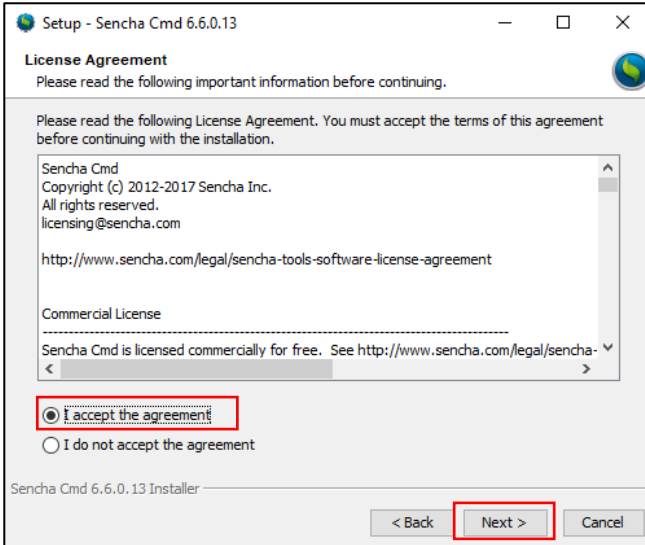
Gambar 2. Sencha Cmd Type Windows

Double click installer, maka akan muncul *command box* seperti gambar di bawah ini. Pilih next.

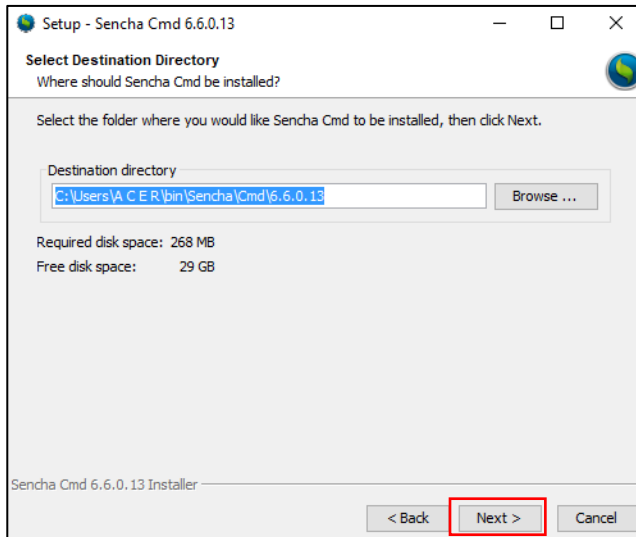


Gambar 3. Sencha Cmd Setup Wizard



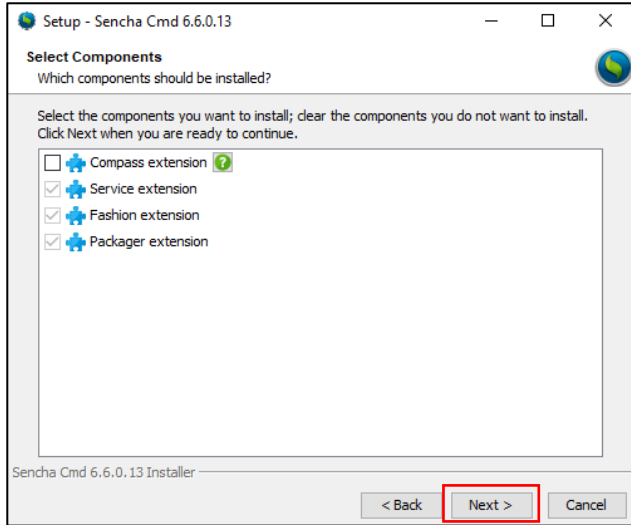


Gambar 4. Sencha Cmd Setup License Agreement

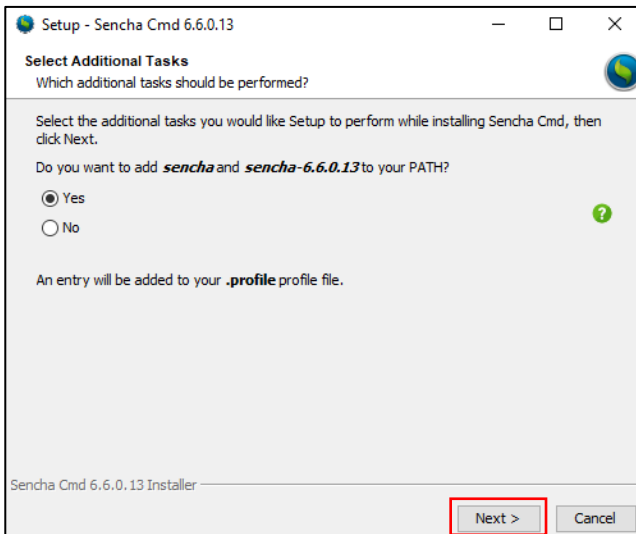


Gambar 5. Sencha Cmd Setup Select Directory



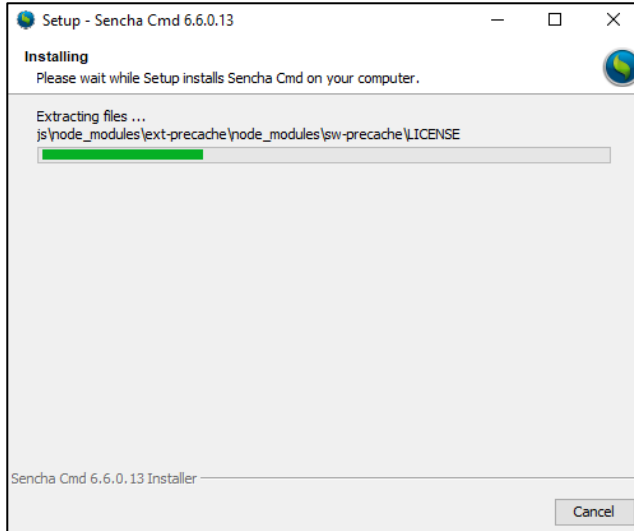


Gambar 6. Sencha Cmd Select Components

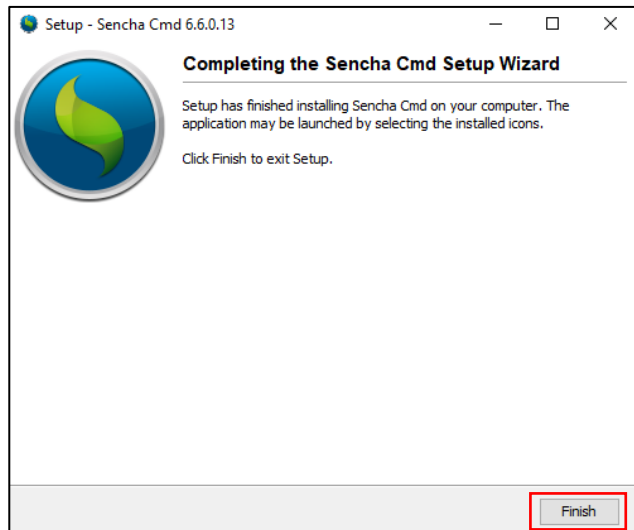


Gambar 7. Select Additional Tasks





Gambar 8. Installing

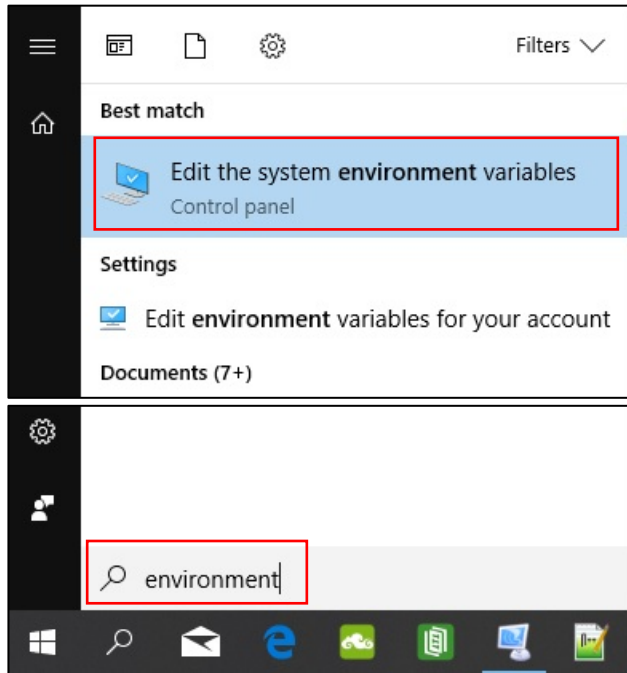


Gambar 9. Completing the sencha cmd setup wizard



Setelah sencha cmd berhasil diinstall, bagi pengguna Windows, tambahkan path pada *Environment Variable* sebagai berikut.

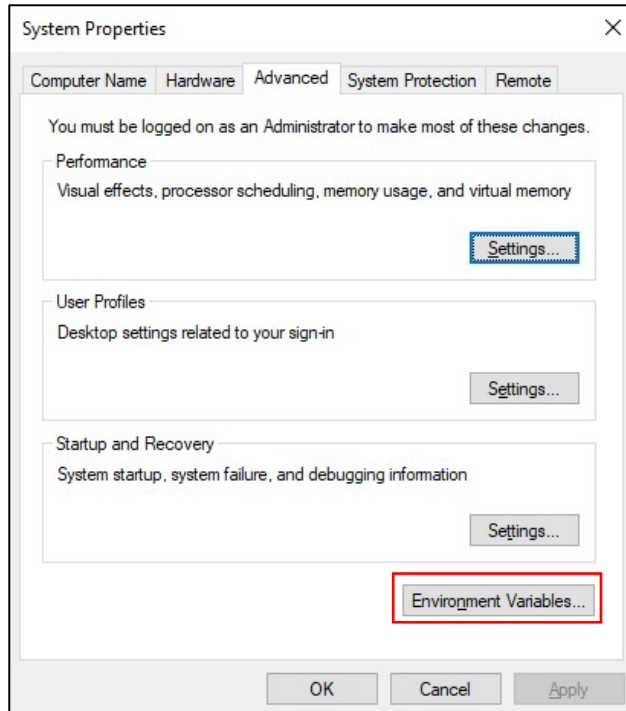
⇒ Browse *Environment* maka akan muncul *Edit the System Environment Variables*. klik menu tersebut.



Gambar 10. Browse Windows

⇒ Pada *System Properties* pilih *Environment Variable*.

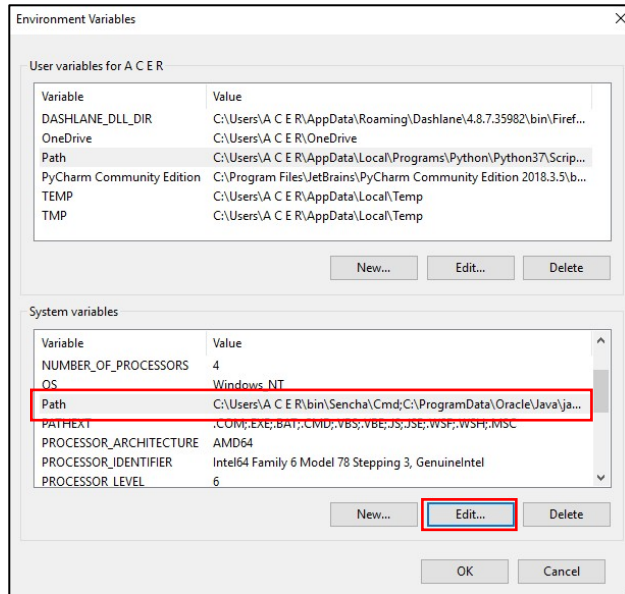




Gambar 11. System Properties

⇒ Pada *System Variables*, pilih *Path* dan Lalu klik tombol *Edit*.



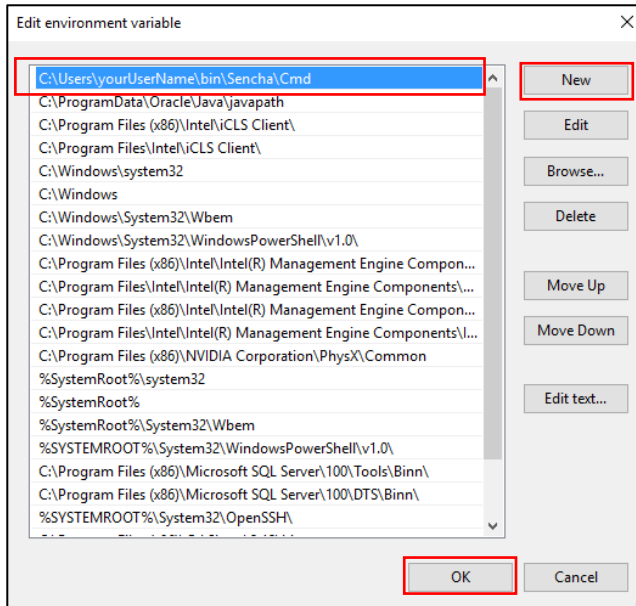


Gambar 12. Environment variables

Pada “edit environment variables”, klik tombol “new”, lalu tambahkan path sebagai berikut.

```
C:\Users\yourUserName\bin\Sencha\Cmd
```





Gambar 13. Edit Environment Variables

Untuk memastikan Sencha Cmd telah terinstall secara sempurna, buka command prompt lalu ketikkan command berikut (di Windows):

```
sencha
```

Jika Sencha Cmd terinstall secara sempurna, informasi mengenai versi Sencha Cmd yang terinstall akan ditampilkan.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\A C E R\sencha
Sencha Cmd v6.6.0.13
Sencha Cmd provides several categories of commands and some global
switches. In
most cases, the first step is to generate an application based on
a Sencha SDK
such as Ext JS or Sencha Touch:

    sencha -sdk /path/to/sdk generate app MyApp /path/to/myapp

Sencha Cmd supports Ext JS 4.1.1a and higher and Sencha Touch 2.1
and higher.

To get help on commands use the help command:

    sencha help generate app

For more information on using Sencha Cmd, consult the guides found
here:

http://docs.sencha.com/cmd/
```

Gambar 14. Sencha Cmd Version

2. Download the Ext JS 6.2.0 SDK dari tautan berikut:

<https://www.sencha.com/legal/GPL>.

Sebelum ekstrak ext js lakukan terlebih dahulu membuat folder sencha-sdks di user pc.

```
mkdir sencha-sdks
```

Setelah itu ekstrak Ext JS di dalam direktori “sencha-sdks” pada “Users” PC Anda. Kemudian anda akan melihat sub-folder seperti berikut (di Windows):

```
C:\Users\Me\sencha-sdks\ext-6.2.0
```



```
C:\WINDOWS\system32\cmd.exe
C:\Users\A C E R>cd sencha-sdks/ext-6.2.0
C:\Users\A C E R\sencha-sdks\ext-6.2.0>dir
Volume in drive C is Acer
Volume Serial Number is 6629-43DE

Directory of C:\Users\A C E R\sencha-sdks\ext-6.2.0

09/09/2019  20.34    <DIR>          .
09/09/2019  20.34    <DIR>          ..
31/08/2016  14.42    <DIR>          .sencha
31/08/2016  14.50    <DIR>          build
31/08/2016  10.03           24.400 build.xml
31/08/2016  10.03    <DIR>          classic
31/08/2016  14.43    <DIR>          cmd
09/09/2019  20.34    <DIR>          docs
31/08/2016  14.49    <DIR>          examples
31/08/2016  10.03           2.370 ext-bootstrap.js
31/08/2016  10.03           548 index.html
31/08/2016  14.49           2.541 LICENSE
31/08/2016  14.49    <DIR>          licenses
31/08/2016  10.03    <DIR>          modern
31/08/2016  14.49           1.481 package.json
31/08/2016  10.03    <DIR>          packages
31/08/2016  10.03            87 Readme.md
31/08/2016  10.42           846.356 release-notes.html
31/08/2016  10.03    <DIR>          resources
31/08/2016  14.49    <DIR>          sass
31/08/2016  10.03    <DIR>          templates
31/08/2016  11.02           273 version.properties
31/08/2016  14.50    <DIR>          welcome
           8 File(s)          878.056 bytes
          15 Dir(s)    31.239.098.368 bytes free
```

Gambar 15. Sub folder Ext Js

Aktifkan sencha web untuk dapat mengakses sencha sdk server di <http://localhost:1841> dengan perintah berikut:

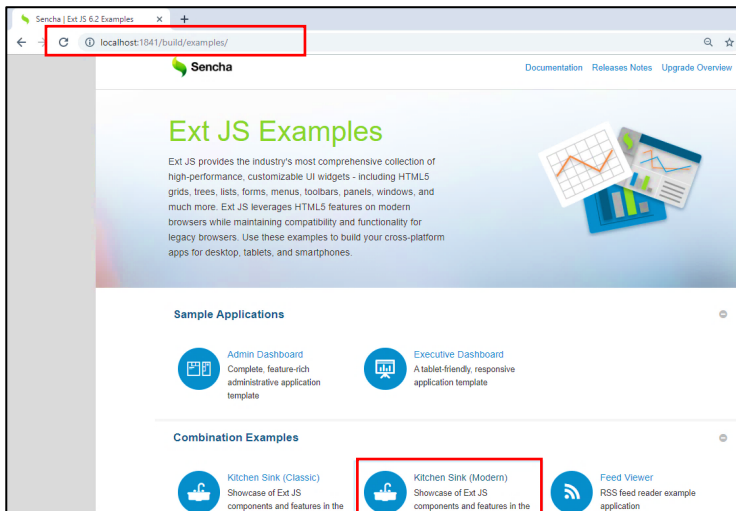
```
sencha web start
```



```
C:\WINDOWS\system32\cmd.exe - sencha web start
C:\Users\A C E R\sencha-sdks\ext-6.2.0>sencha web start
Sencha Cmd v6.6.0.13
[INF] Starting server on port : 1841
[INF] Mapping http://localhost:1841/ to C:\Users\A C E R\
sencha-sdks\ext-6.2.0...
[INF] Server started at port : 1841
```

Gambar 16. Sencha Web Start

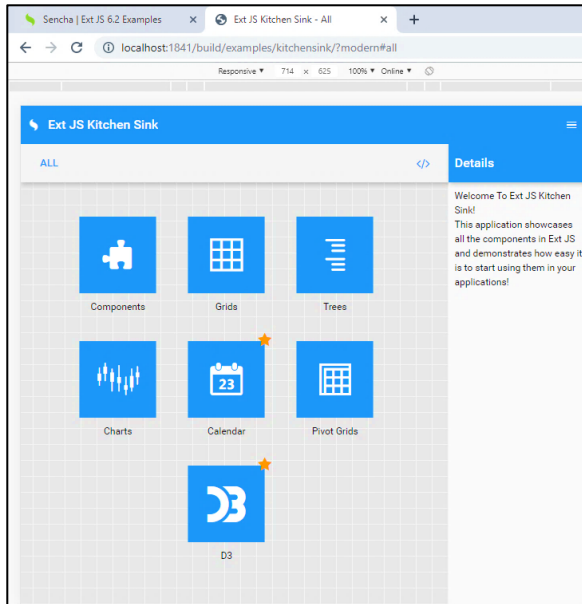
Ketikkan localhost:1841 di browser app, kemudian Pilih Kitchen Sink (modern) pada Combination Example, seperti gambar dibawah ini.



Gambar 17. Sencha Ext JS 6.2.0 Example

Setelah Kitchen Sink (modern) dipilih akan menampilkan semua komponen di Ext JS Kitchen Sink.





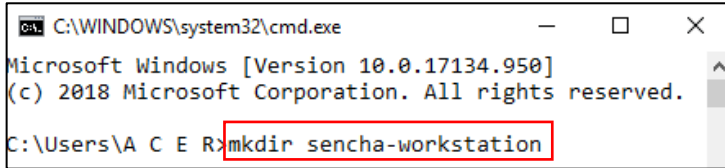
Gambar 18. Ext JS Kitchen Sink

Workstation

Untuk memudahkan Anda dalam pembangunan aplikasi dengan sencha secara terstruktur, buatlah folder sencha-workstation sebagai folder kerja Anda. Buat direktori kosong satu level dengan sencha-sdks yang telah dibuat sebelumnya dengan perintah di *command prompt*:

```
mkdir sencha-workstation
```





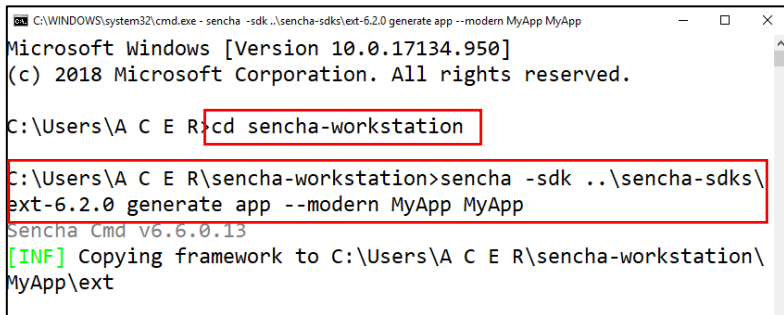
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\A C E R>mkdir sencha-workstation
```

Gambar 19. Pembuatan Folder Sencha Workstation

Membuat Kerangka Aplikasi

Untuk membuat kerangka aplikasi, masukkan perintah berikut pada *command prompt* Anda:

```
cd C:\Users\Me\sencha-workstation
sencha -sdk ..\sencha-sdks\ext-6.2.0 generate app --modern MyApp
MyApp
```



```
C:\WINDOWS\system32\cmd.exe - sencha -sdk ..\sencha-sdks\ext-6.2.0 generate app --modern MyApp MyApp
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\A C E R>cd sencha-workstation
C:\Users\A C E R\sencha-workstation>sencha -sdk ..\sencha-sdks\
ext-6.2.0 generate app --modern MyApp MyApp
Sencha Cmd v6.6.0.13
[INF] Copying framework to C:\Users\A C E R\sencha-workstation\
MyApp\ext
```

Gambar 20. Pembuatan Kerangka Aplikasi Ext Js

Maka struktur aplikasi Ext Js akan tersalin ke dalam direktori “MyApp”.

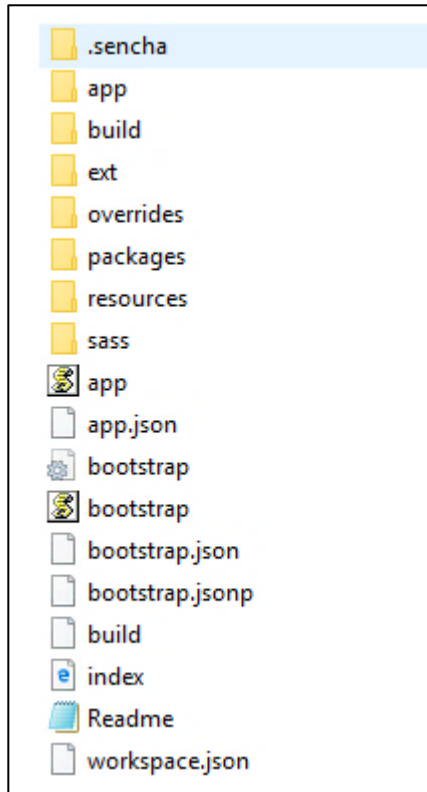
Kerangka aplikasi Ext JS yang dibuat dalam direktori “MyApp” memiliki struktur direktori yang sama untuk



setiap aplikasi seperti yang terlihat pada Gambar 21. Untuk memudahkan menggunakan aplikasi, direkomendasikan semua kelas *Store*, *Model*, *ViewModel*, dan *ViewController* ditempatkan di folder *app* dengan Model ditempatkan dalam folder *model*, *Store* di folder *store*, dan *ViewModel/Controller* dalam folder *view*.

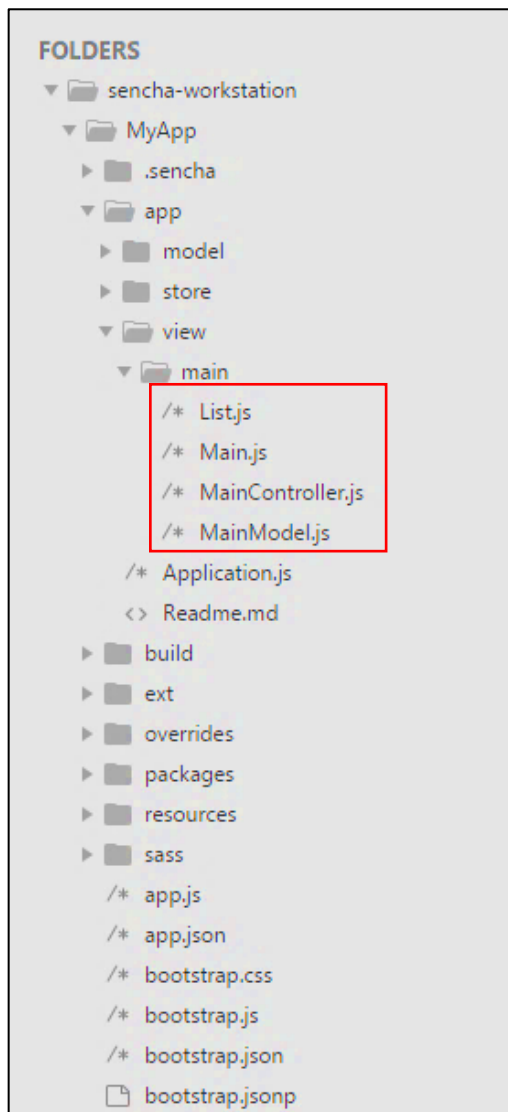
ViewController dan *ViewModel* dikelompokkan bersama dalam sub-folder dalam folder *app/view/main* menggunakan struktur penamaan yang sama dengan nama *View* yang akan Anda gunakan. Seperti yang terlihat pada Gambar 22, folder *ViewModel/Controller* diberi nama *MainModel.js/MainController.js*.





Gambar 21. Kerangka Aplikasi "MyApp"





Gambar 22. Struktur Direktori "MyApp"



Menjalankan Aplikasi

Pembangunan aplikasi sencha dimonitori oleh browser, untuk mengakses aplikasi yang dihasilkan melalui server lokal, lakukan perintah berikut ini dari direktori yang sama (C:\Users\Me\sencha-workstation\MyApp):

```
sencha app watch
```

Pada *command prompt*, Anda akan melihat beberapa baris informasi ketika *library* aplikasi Sencha Cmd dimuat, ada dua yang perlu diperhatikan untuk memastikan aplikasi sudah berhasil dimuat.

```
...  
[INF] Application available at http://localhost:1841  
...  
[INF] Waiting for changes...
```





Gambar 23. Aplikasi dan Interface Sencha Cmd

Sekarang Anda dapat mengetikkan `http://localhost:1841` pada browser dan melihat aplikasi sencha yang telah Anda buat.

Localhost ini menyediakan tampilan muka (*interface*) Sencha Cmd untuk memonitoring aktifitas yang dilakukan oleh User seperti perubahan pada styling atau kode JavaScript. Sencha Cmd akan mendeteksi perubahan ini dan memperbarui output build.

Ketika perubahan terdeteksi, Sencha Cmd akan mencatat beberapa baris output dan diakhiri dengan notifikasi "Waiting for changes ..." (Menunggu perubahan



...)" untuk memberi tahu Anda bahwa semuanya sudah siap.

Proses perubahan kode pada aplikasi dilakukan dengan menggunakan IDE atau editor teks seperti *Sublime*. Jika sudah selesai melakukan penyuntingan, simpan perubahan (Ctrl+ S), lalu muat kembali pada (Ctrl + R) browser untuk melihat apakah kode sudah sesuai dengan ekspektasi Anda. Anda dapat menghentikan simulasi aplikasi dengan menekan CTRL + C.

Latihan

1. Pelajari perbedaan aplikasi yang dihasilkan menggunakan kode **sencha app init** dan **sencha generate app**
2. Buat aplikasi sederhana dengan nama sendiri (selain MyApp).

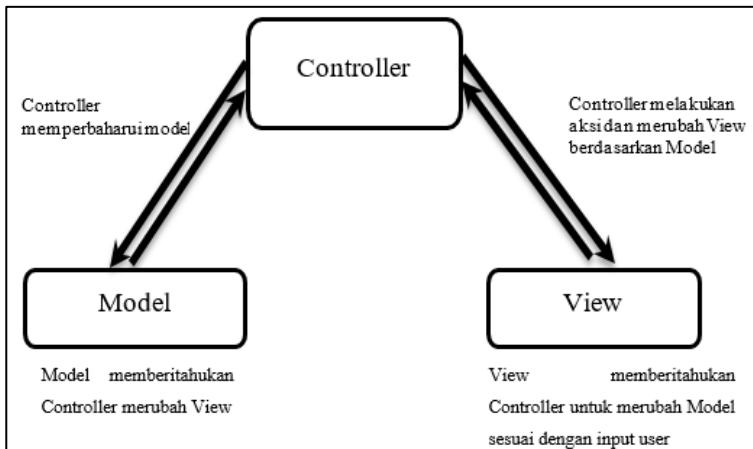


2. MODEL, VIEW, CONTROLLER

Pengenalan MVC

MVC merupakan singkatan dari *Model View Controller*. MVC membagi aplikasi sencha mejadi beberapa komponen berdasarkan arsitektur logika program untuk memudahkan pengaturan aplikasi.

Diagram di bawah ini menggambarkan arsitektur MVC (Sencha Touch - MVC, 2019).



Gambar 24. MVC : Model View Controller



Dalam arsitektur MVC

- *Model* merepresentasikan bentuk object yang berfungsi sebagai penghubung ke database dan memberitahukan Controller untuk merubah View.
- *Views* merupakan visualisasi yang dilihat oleh user. View berperan sebagai tampilan muka (*interface*) dari aplikasi dan memberitahu *Controller* merubah Model sesuai input dari user.
- *Controller* sebagai pengendali keseluruhan aplikasi. Secara sederhana, data dari input user disimpan dalam *Model*, interaksi tersebut dimonitori oleh *Controller* yang kemudian merespon interaksi dengan memperbarui *View* dan *Model*, jika diperlukan.

Seperti yang terlihat pada Gambar 8, *View* dan *Model* tidak terhubung satu sama lain, hal ini dikarenakan *Controller* memiliki tanggung jawab tunggal melakukan pembaruan kode.

Tujuan dari MVC adalah untuk secara jelas mendefinisikan tanggung jawab setiap kelas dalam aplikasi sehingga membuat aplikasi lebih mudah diuji, dipelihara, dan memudahkan kodenya untuk digunakan kembali.

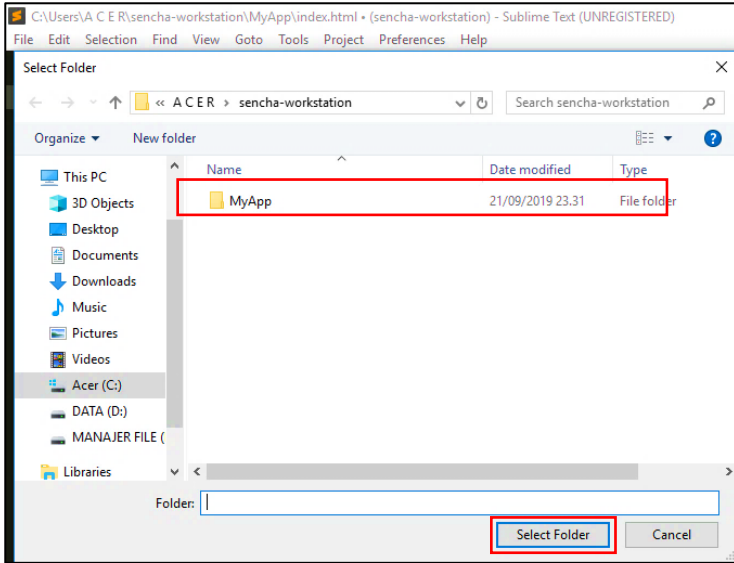


Membuka Kode Aplikasi dengan Sublime Text

Sublime merupakan editor teks yang banyak digunakan. Salah satu alasan Sublime menjadi pilihan yaitu tampilan Sublime membedakan kode menjadi beberapa warna tertentu sesuai dengan pengelompokan kode. Untuk memulai membangun aplikasi, bukanlah kode aplikasi menggunakan Sublime dengan mengikuti langkah- langkah berikut ini.

1. Apabila Sublime tersedia di PC Anda, download dan install software Sublime Text terlebih dahulu:
<https://www.sublimetext.com/3>
2. Buka software Sublime Text, klik File → Open Folder
3. Pilih folder aplikasi MyApp yang telah dibuat, klik Select Folder





Gambar 25. Folder “MyApp”

index.html

Mari kita mulai lihat kode index.html terlebih dahulu. Ext JS menggunakan Microloader untuk memuat kode-kode aplikasi yang dijelaskan dalam file app.json. Dengan cara ini, Anda tidak perlu lagi menambahkan kode-kode aplikasi tersebut secara manual ke index.html. Dengan app.json semua aplikasi metadata ada di satu lokasi. Sencha Cmd kemudian dapat mengkompilasi aplikasi Anda dengan cara yang sederhana dan efisien.



```
1 <!DOCTYPE HTML>
2 <html manifest="">
3 <head>
4   <meta http-equiv="X-UA-Compatible" content="IE=edge">
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width,
7     initial-scale=1, maximum-scale=10, user-scalable=yes">
8   <title>MyApp</title>
9
10  |
11  <script id="microloader" data-app="
12    821fbc87-9474-4dda-8b80-a7d693901087" type="text/
13    javascript" src="bootstrap.js"></script>
14 </head>
15 <body></body>
16 </html>
```

Gambar 26. Index.html

Application.js

Setiap aplikasi Ext JS dimulai dengan sebuah instance dari kelas `Application`. Kelas ini dimaksudkan agar dapat diluncurkan oleh `app.js`. Konten `Application.js` berikut secara otomatis dibuat ketika Anda membuat aplikasi Anda dengan `Sencha Cmd`.



```

Ext.define('MyApp.Application', {
    extend: 'Ext.app.Application',

    name: 'MyApp',

    stores: [
        // TODO: add global / shared stores here
    ],

    launch: function () {
        // TODO - Launch the application
    },

    onAppUpdate: function () {
        Ext.Msg.confirm('Application Update', 'This application has an
update, reload?',
            function (choice) {
                if (choice === 'yes') {
                    window.location.reload();
                }
            }
        );
    }
});

```

app.js

Dalam `app.js`, kelas `Application` (`MyApp.Application`) diluncurkan. Dengan menetapkan kelas kontainer untuk `mainView`, Anda dapat menggunakan kelas apa pun sebagai area pandang (*Viewport*) Anda. Dalam contoh di atas, telah ditentukan `MyApp.view.main.Main` (kelas `TabPanel`) untuk menjadi *Viewport* pada aplikasi *MyApp* ini.



Konfigurasi `mainView` menginstruksikan aplikasi untuk membuat *View* yang ditentukan dan melampirkan Plugin `Viewport`. Ini menghubungkan *View* ke badan dokumen html.

```
Ext.application({
  name: 'MyApp',
  extend: 'MyApp.Application',
  requires: [
    'MyApp.view.main.Main'
  ],
  mainView: 'MyApp.view.main.Main'
});
```

View

View merupakan suatu komponen yang merupakan subkelas dari `Ext.Component`. Sebuah tampilan berisi semua aspek visual aplikasi Anda.

Jika Anda membuka file `app/view/main/Main.js`, Anda akan melihat kode berikut.



```

Ext.define('MyApp.view.main.Main', {
    extend: 'Ext.tab.Panel',
    xtype: 'app-main',

    requires: [
        'Ext.MessageBox',
        'MyApp.view.main.MainController',
        'MyApp.view.main.MainModel',
        'MyApp.view.main.List'
    ],

    controller: 'main',
    viewModel: 'main',

    defaults: {
        tab: {
            iconAlign: 'top'
        },
        styleHtmlContent: true
    },

    tabBarPosition: 'bottom',

    items: [
        {
            title: 'Home',
            iconCls: 'x-fa fa-home',
            layout: 'fit',
            items: [{
                xtype: 'mainlist'
            }]
        }, {
            title: 'Users',
            iconCls: 'x-fa fa-user',
            bind: {

```




```

        html: '{loremIpsum}'
    }
}, {
    title: 'Groups',
    iconCls: 'x-fa fa-users',
    bind: {
        html: '{loremIpsum}'
    }
}, {
    title: 'Settings',
    iconCls: 'x-fa fa-cog',
    bind: {
        html: '{loremIpsum}'
    }
}
]
});

```

Harap dicatat bahwa *View* tidak termasuk logika aplikasi apa pun. Semua logika *View* harus dimasukkan dalam *ViewController*, yang akan kita bicarakan di bagian selanjutnya.

Dua bagian menarik dari *View* ini adalah konfigurasi controller dan *viewModel*.

Konfigurasi *controller* memungkinkan Anda untuk menentukan *ViewController* untuk *View*. Ketika *ViewController* ditentukan pada *View* dengan cara ini, *ViewController* kemudian menjadi wadah untuk penanganan dan referensi *event* yang ditangkap dari interaksi pengguna



dengan komponen pada *View*.

Konfigurasi *viewModel* memungkinkan Anda untuk menetapkan *ViewModel* untuk *View*. *ViewModel* adalah penyedia data untuk komponen ini dan *View* turunannya. Data yang terdapat dalam *ViewModel* biasanya digunakan dengan menambahkan konfigurasi *bind* ke komponen yang ingin menyajikan atau mengedit data ini.

Dalam *View* "Main", Anda dapat melihat bahwa judul tajuk tabpanel utama terikat ke *ViewModel*. Ini berarti bahwa judul akan diisi oleh nilai "name" dari data, yang dikelola dalam *ViewModel*. Jika data *ViewModel* berubah, nilai judul akan diperbarui secara otomatis.

Tampilan berikutnya yang menarik adalah tampilan "List" di `app/view/main/List.js`. *View* ini dipanggil pada `Main.js` dengan mereferensikan `xtype: 'mainlist'`. Dengan menetapkan konfigurasi *listeners select* pada *Grid* (Tabel), setiap data yang dipilih (dengan tap/sentuhan) akan terhubung dengan fungsi 'onItemSelected' yang dikelola oleh *ViewController*.



```

Ext.define('MyApp.view.main.List', {
    extend: 'Ext.grid.Grid',
    xtype: 'mainlist',

    requires: [
        'MyApp.store.Personnel'
    ],

    title: 'Personnel',

    store: {
        type: 'personnel'
    },

    columns: [
        { text: 'Name', dataIndex: 'name', width: 100 },
        { text: 'Email', dataIndex: 'email', width: 230 },
        { text: 'Phone', dataIndex: 'phone', width: 150 }
    ],

    listeners: {
        select: 'onItemSelected'
    }
});

```

View Controller (MainController)

Jika anda membuka file app/view/main/ MainController.js ,
Anda akan melihat kode berikut.



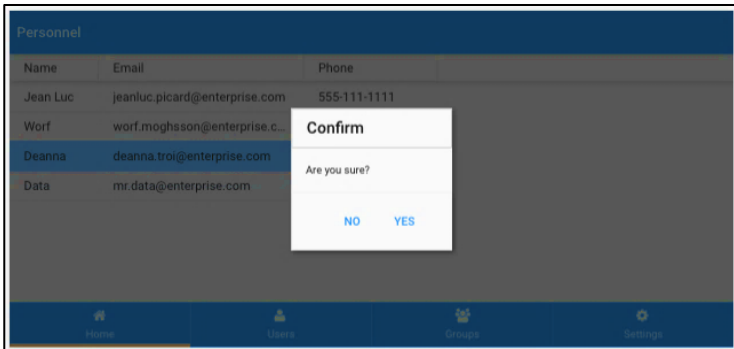
```
Ext.define('MyApp.view.main.MainController', {
    extend: 'Ext.app.ViewController',

    alias: 'controller.main',

    onItemSelected: function (sender, record) {
        Ext.Msg.confirm('Confirm', 'Are you sure?', 'onConfirm', this);
    },

    onConfirm: function (choice) {
        if (choice === 'yes') {
            //
        }
    }
});
```

Perhatikan bahwa pada ViewController ini, fungsi ‘onItemSelected’ akan membuat komponen pesan konfirmasi yang akan muncul pada layar seperti berikut.



Gambar 27. Pesan Konfirmasi



View Model (MainModel)

Jika Anda membuka file `app/view/main/MainModel.js`, Anda akan melihat kode berikut. Perhatikan bahwa data `loremIpsum` yang ditemukan pada `Main.js` didefinisikan di *ViewModel* ini.

```
Ext.define('MyApp.view.main.MainModel', {
    extend: 'Ext.app.ViewModel',

    alias: 'viewmodel.main',

    data: {
        name: 'MyApp',

        loremIpsum: 'Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore
magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.'
    }
    //TODO - add data, formulas and/or methods to support your view
});
```

Store

Jika Anda membuka file `app/store/Personnel.js`, Anda akan melihat kode berikut. *Store* adalah data di sisi klien (merujuk ke definisi data di kelas *Model*). *Store* menyediakan fungsi untuk menyortir, memfilter dan



memuat data yang ada di dalamnya. Perhatikan bahwa Grid di View List.js mengambil data dari *Store* ini. Dalam contoh ini, *Store* secara langsung berisi data lokal. Sebagian besar situasi dunia nyata mengharuskan Anda mengumpulkan data dengan menggunakan *proxy* pada *Model* atau *Store* Anda. *Proxy* memungkinkan transfer data antara penyedia data dan aplikasi Anda.



```

Ext.define('MyApp.store.Personnel', {
    extend: 'Ext.data.Store',

    alias: 'store.personnel',

    fields: [
        'name', 'email', 'phone'
    ],
    data: {
        items: [
            {
                name: 'Jean Luc',
                email: "jeanluc.picard@enterprise.com",
                phone: "555-111-1111"
            },
            {
                name: 'Worf',
                email: "worf.moghsson@enterprise.com",
                phone: "555-222-2222"
            },
            {
                name: 'Deanna',
                email: "deanna.troi@enterprise.com",
                phone: "555-333-3333"
            },
            {
                name: 'Data',
                email: "mr.data@enterprise.com",
                phone: "555-444-4444"
            }
        ]
    },
    proxy: {
        type: 'memory',
        reader: {
            type: 'json',
            rootProperty: 'items'
        }
    }
});

```



Latihan

1. Buat aplikasi sederhana lengkap dengan unsur View, View Controller, View Model, dan Store.
2. Apa keunggulan arsitektur MVC yang Anda rasakan setelah membuat aplikasi sederhana tersebut?



3. KOMPONEN ANTARMUKA PENGGUNA

Layout and Container

Sistem tata letak / *layout* adalah salah satu bagian Ext JS yang paling kuat. Sistem tata letak ini menangani ukuran dan posisi setiap komponen dalam aplikasi Anda. *Container* adalah jenis komponen khusus yang dapat mengandung komponen lain. Aplikasi Ext JS yang khas terdiri dari beberapa lapis komponen bertumpuk.

Buat file baru berikut: `app/view/main/Panel.js`.

```
Ext.define('MyApp.view.main.Panel', {
    extend: 'Ext.Container',
    xtype: 'panel-basic',

    requires: [
        'Ext.layout.VBox'
    ],
    layout: {
        type: 'vbox',
        pack: 'center',
        align: 'stretch'
    },

    defaults: {
        bodyPadding: 10
    },
});
```



```

items: [{
  layout: {
    type: 'hbox',
    pack: 'center',
    align: 'stretch'
  },
  autoSize: true,
  defaults: {
    flex: 1,
    bodyPadding: 10,
    autoSize: true,
    html: '{loremIpsum}'
  },
  items: [{
    xtype: 'panel',
    shadow: 'true',
    margin: 10
  }, {
    xtype: 'panel',
    shadow: 'true',
    title: 'Title',
    margin: 10
  }]
}],
{
  layout: {
    type: 'hbox',
    pack: 'center',
    align: 'stretch'
  },
  autoSize: true,
  defaults: {
    flex: 1,
    bodyPadding: 10,
    autoSize: true,
    html: '{loremIpsum}'
  },
}

```



```

items: [{
  xtype: 'panel',
  shadow: 'true',
  margin: 10,
  title: 'Built in Tools',
  html: '{loremIpsum}',
  tools: [
    {type: 'minimize'},
    {type: 'refresh'},
    {type: 'search'},
    {type: 'save'},
    {type: 'menu'}
  ]
}, {
  xtype: 'panel',
  shadow: 'true',
  margin: 10,
  title: 'Custom Tools using iconCls',
  html: '{loremIpsum}',
  autoSize: true,
  tools: [
    {iconCls: 'x-fa fa-wrench'},
    {iconCls: 'x-fa fa-reply'},
    {iconCls: 'x-fa fa-reply-all'}
  ]
}]
});

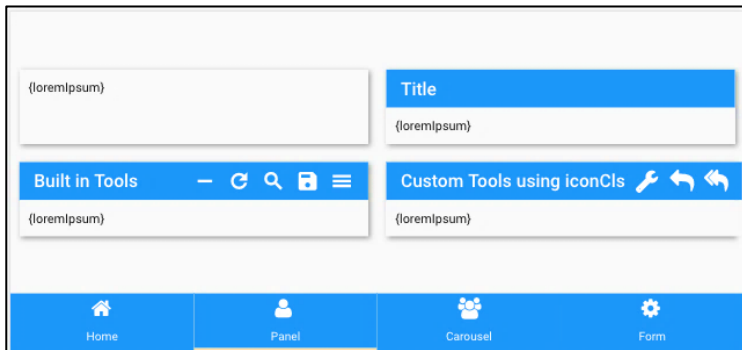
```

Silakan integrasikan kode berikut sebagai item pada `MyApp.view.main.Main` menggunakan referensi `xtype: 'panel-basic'`.



```
{
  title: 'Panel',
  iconCls: 'x-fa fa-user',
  layout: 'fit',
  items: [{
    xtype: 'panel-basic'
  }]
}
```

Berikut tampilan Panel tersebut:



Gambar 28. Panel

Carousel

Carousel adalah cara yang bagus untuk memungkinkan pengguna menggesek (*swipe*) melalui beberapa halaman layar penuh. Carousel hanya menampilkan salah satu halamannya pada satu waktu, tetapi memungkinkan Anda menelusuri halaman lain menggunakan gerakan menggesek. Anda dapat menganggap Carousel sebagai item yang



aktif, dengan sisa item yang membentang ke kiri dan kanan. Titik indikator memvisualisasikan berapa banyak layar yang tersedia untuk digesek, seperti yang ditunjukkan pada gambar berikut:



Gambar 29. Carousel

Buat file baru berikut: `app/view/main/Carousel.js`.

```
Ext.define('MyApp.view.main.Carousel',
{
    extend: 'Ext.carousel.Carousel',
    xtype: 'carousel-basic',

    defaults: {
        flex: 1
    },
    items: [
        {
            xtype: 'list',
            items: {
                xtype: 'toolbar', docked: 'top', title: 'Roster'
            },
            store: {
                fields: ['name'],
```



```

        data: [
            {name: 'Jon'},
            {name: 'Sansa'},
            {name: 'Arya'},
            {name: 'Robb'},
            {name: 'Bran'},
            {name: 'Rickon'}
        ]
    },
    itemTpl: '{name}'
},
{
    xtype: 'fieldset',
    items: [
        {
            xtype: 'toolbar',
            docked: 'top',
            title: 'Login'
        },
        {
            xtype: 'textfield',
            label: 'Name'
        },
        {
            xtype: 'passwordfield',
            label: 'Password'
        }
    ]
}
].
});

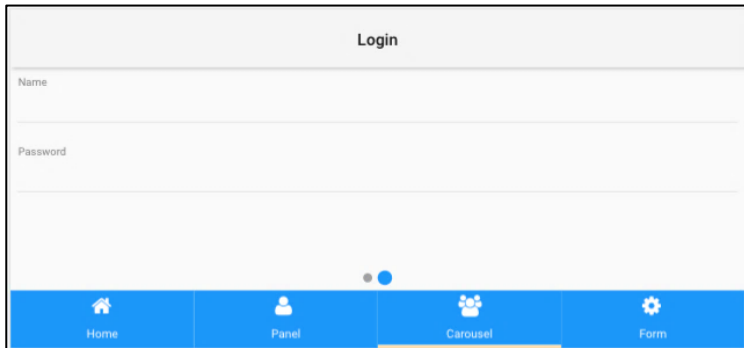
```

Silakan integrasikan kode berikut sebagai item Carousel pada MyApp.view.main.Main menggunakan referensi xtype: 'carousel-basic'.



```
{
  title: 'Carousel',
  iconCls: 'x-fa fa-user',
  layout: 'fit',
  items: [{
    xtype: 'carousel-basic'
  }]
}
```

Berikut tampilan Carousel tersebut:



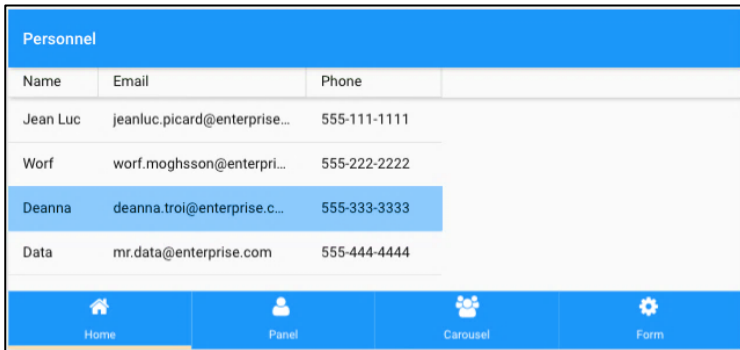
Gambar 30. Carousel “Login”

Grids

Komponen ini memaparkan data dalam bentuk tabel dengan memungkinkan pemaparan beberapa kolom. Komponen ini telah dibahas di BAB 2 pada bagian `app/view/main/List.js`.



Berikut tampilan Grid tersebut:



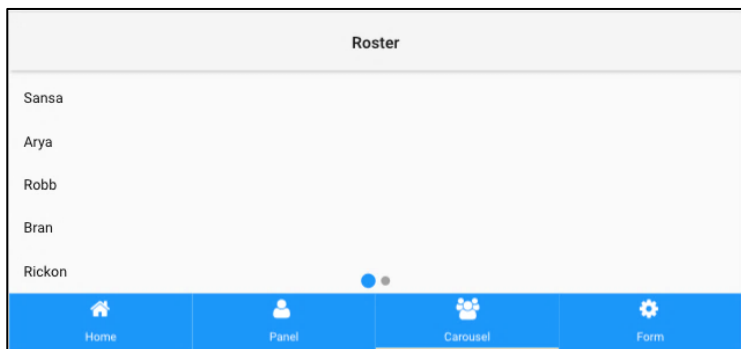
| Personnel | | | |
|-----------|------------------------------|--------------|--|
| Name | Email | Phone | |
| Jean Luc | jeanluc.picard@enterprise... | 555-111-1111 | |
| Worf | worf.moghsson@enterpri... | 555-222-2222 | |
| Deanna | deanna.troi@enterprise.c... | 555-333-3333 | |
| Data | mr.data@enterprise.com | 555-444-4444 | |

Gambar 31. List Grid

List

Komponen ini memaparkan data dalam bentuk list. Komponen ini memiliki xtype: 'list' dan sudah dipaparkan sebagai salah satu item di Carousel yang telah dijelaskan sebelumnya.

Berikut tampilan List pada Carousel di atas:



Gambar 32. Carousel "List"



Forms

Panel Forms tidak lebih dari Panel dasar dengan kemampuan penanganan formulir seperti validasi data dan mendapatkan seluruh nilai dari komponen isian form tersebut. Panel Forms dapat digunakan di seluruh aplikasi Ext di mana pun ada kebutuhan untuk mengumpulkan data dari pengguna.

Selain itu, Panel Forms dapat menggunakan Container Layout apa pun, menyediakan cara yang mudah dan fleksibel untuk menangani pemosisian bidangnya. Panel Forms juga dapat diikat ke Model, membuatnya mudah untuk memuat data dari dan mengirimkan data kembali ke server.

Panel Forms membungkus formulir dasar yang menangani semua pengelolaan layanan masukan/*input*, validasi, pengiriman, dan pemuatan formulir. Ini berarti bahwa banyak opsi konfigurasi dari formulir dasar dapat digunakan langsung pada Panel Forms.

Buat file baru berikut: `app/view/main/Form.js`.



```

Ext.define('MyApp.view.main.Form', {
    extend: 'Ext.form.Panel',
    xtype: 'form-basic',
    requires: [
        'Ext.form.FieldSet',
        'Ext.field.Number',
        'Ext.field.Spinner',
        'Ext.field.Password',
        'Ext.field.Email',
        'Ext.field.Url',
        'Ext.field.DatePicker',
        'Ext.field.Select',
        'Ext.field.Hidden',
        'Ext.field.Radio'
    ],
    shadow: true,
    cls: 'demo-solid-background',
    id: 'basicform',
    items: [
        {
            xtype: 'fieldset',
            id: 'fieldset1',
            title: 'Personal Info',
            instructions: 'Please enter the information above.',
            defaults: {
                labelWidth: '35%'
            },
            items: [
                {
                    xtype: 'textfield',
                    name: 'name',
                    label: 'Name',
                    placeHolder: 'Tom Roy',
                    autoCapitalize: true,
                    required: true,
                    clearIcon: true
                },
                {
                    xtype: 'passwordfield',

```



```

    revealable: true,
    name : 'password',
    label: 'Password',
    clearIcon: true
  },
  {
    xtype: 'emailfield',
    name: 'email',
    label: 'Email',
    placeholder: 'me@sencha.com',
    clearIcon: true
  },
  {
    xtype: 'urlfield',
    name: 'url',
    label: 'Url',
    placeholder: 'http://sencha.com',
    clearIcon: true
  },
  {
    xtype: 'spinnerfield',
    name: 'spinner',
    label: 'Spinner',
    minValue: 0,
    maxValue: 10,
    clearable: true,
    stepValue: 1,
    cycle: true
  },
  {
    xtype: 'checkboxfield',
    name: 'cool',
    label: 'Cool',
    platformConfig: {
      '!desktop': {
        bodyAlign: 'end'
      }
    }
  }
},

```



```

{
    xtype: 'datepickerfield',
    destroyPickerOnHide: true,
    name: 'date',
    label: 'Start Date',
    value: new Date(),
    picker: {
        yearFrom: 1990
    }
},
{
    xtype: 'selectfield',
    name: 'rank',
    label: 'Rank',
    options: [
        {
            text: 'Master',
            value: 'master'
        },
        {
            text: 'Journeyman',
            value: 'journeyman'
        },
        {
            text: 'Apprentice',
            value: 'apprentice'
        }
    ]
},
{
    xtype: 'sliderfield',
    name: 'slider',
    label: 'Slider'
},
{
    xtype: 'togglefield',
    name: 'toggle',
    label: 'Toggle'
},

```



```

        {
            xtype: 'textareafield',
            name: 'bio',
            label: 'Bio'
        }
    ]
},
{
    xtype: 'fieldset',
    id: 'fieldset2',
    title: 'Favorite color',
    platformConfig: {
        '!desktop': {
            defaults: {
                bodyAlign: 'end'
            }
        }
    },
    defaults: {
        xtype: 'radiofield',
        labelWidth: '35%'
    },
    items: [
        {
            name: 'color',
            value: 'red',
            label: 'Red'
        },
        {
            name: 'color',
            label: 'Blue',
            value: 'blue'
        },
        {
            name: 'color',
            label: 'Green',
            value: 'green'
        }
    ],

```



```

        {
            name: 'color',
            label: 'Purple',
            value: 'purple'
        }
    ]
},
{
    xtype: 'container',
    defaults: {
        xtype: 'button',
        style: 'margin: 1em',
        flex: 1
    },
    layout: {
        type: 'hbox'
    },
    items: [
        {
            text: 'Disable fields',
            ui: 'action',
            scope: this,
            hasDisabled: false,
            handler: function(btn){
                var fieldset1 = Ext.getCmp('fieldset1'),
                    fieldset2 = Ext.getCmp('fieldset2');

                if (btn.hasDisabled) {
                    fieldset1.enable();
                    fieldset2.enable();
                    btn.hasDisabled = false;
                    btn.setText('Disable fields');
                } else {
                    fieldset1.disable();
                    fieldset2.disable();
                    btn.hasDisabled = true;
                    btn.setText('Enable fields');
                }
            }
        }
    ],
},

```



```

        {
            text: 'Reset',
            ui: 'action',
            handler: function(){
                Ext.getCmp('basicform').reset();
            }
        }
    ]
}
]);

```

Silakan integrasikan kode berikut sebagai item Carousel pada MyApp.view.main.Main menggunakan referensi xtype: 'form-basic'.

```

{
    title: 'Form',
    iconCls: 'x-fa fa-user',
    layout: 'fit',
    items: [{
        xtype: 'form-basic'
    }]
}

```

Berikut tampilan Form tersebut:



Personal Info

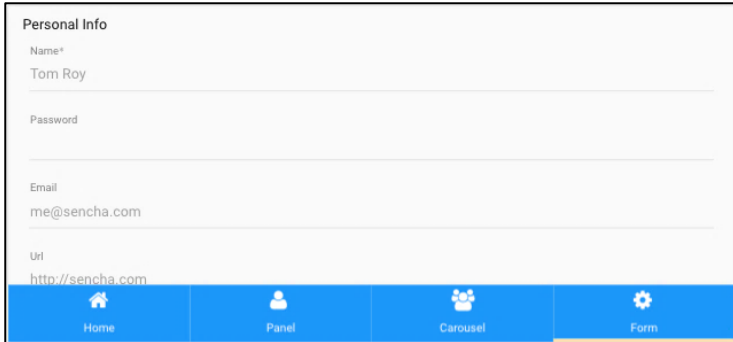
Name*
Tom Roy

Password

Email
me@sencha.com

Url
http://sencha.com

Home Panel Carousel Form



Gambar 33. Forms

Latihan

1. Buat aplikasi sederhana dengan menggunakan minimal 2 antarmuka di atas.



4. PERTUKARAN DATA DENGAN JSON

Pengenalan JSON

JavaScript Object Notation (JSON) merupakan metadata yang menggunakan format yang mudah dibaca untuk melakukan pertukaran data objek yang terdiri dari pasangan atribut-nilai dan tipe data array. *JSON* dengan Padding atau disingkat dengan JSONP merupakan mekanisme untuk melakukan pertukaran data antar domain. Sebagai contohnya, jika aplikasi Sencha Anda sudah di *deploy* dalam bentuk apk Android dan diinstalasi dalam sebuah smartphone Android, maka untuk melakukan pertukaran data ke pangkalan data MySQL melalui skrip PHP yang berada di server, maka kelas JSONP diperlukan.

Pada BAB 2 bagian store Personnel, data statis diambil dan di paparkan dalam atribut data menggunakan tipe proxy memory. Pada BAB 4 ini, kita akan mengambil data dinamis dari pangkalan data MySQL melalui skrip PHP yang berada di http://localhost/MyApp_php/personnel.php. Silakan buat database dengan nama myapp dan buat tabel dengan nama personnel dengan 4 field sebagai berikut:



Tabel 1. Contoh Struktur Tabel Database

| No | Name | Type |
|----|---------|-------------|
| 1 | user_id | int(10) |
| 2 | name | varchar(50) |
| 3 | email | varchar(50) |
| 4 | phone | varchar(20) |

Setelah database selesai dibuat, melalui phpMyAdmin¹, isikan beberapa data ke dalam tabel personnel tersebut, kemudian silakan edit store personnel dengan merubah tipe proxy menjadi jsonp dan menambahkan atribut api – read dengan value berupa alamat skrip personnel php yang akan kita buat.

¹ Download XAMPP di www.apachefriends.org



```
Ext.define('MyApp.store.Personnel', {
    extend: 'Ext.data.Store',
    alias: 'store.personnel',
    storeId: 'personnel',
    autoLoad: true,
    autoSync: true,
    fields: [
        'user_id', 'name', 'email', 'phone'
    ],
    proxy: {
        type: 'jsonp',
        api: {
            read: "http://localhost/MyApp_php/personnel.php"
        }
    },
    reader: {
        type: 'json',
        rootProperty: 'items',
        messageProperty: 'error'
    }
});
```

Berikut detail skrip personnel.php tersebut:



```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myapp";

$conn = new mysqli($servername, $username,
    $password,
    $dbname);
/* check connection */
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

$callback = $_REQUEST['callback'];

// Create the output object.
$output = array();
$success = 'false';
$query="select * from personnel" or die("Cannot access item");
$result = mysqli_query($conn, $query);
if(mysqli_num_rows($result) > 0){
while($obj = mysqli_fetch_object($result)) {
$output[] = $obj;
}
}
$success = 'true';
}
//start output
if ($callback) {
header('Content-Type: text/javascript');
echo $callback . '({"success":' . $success . ',
    "items":' . json_encode($output) . '});';
} else {
header('Content-Type: application/x-json'); echo
json_encode($output);
}
$conn->close();
?>

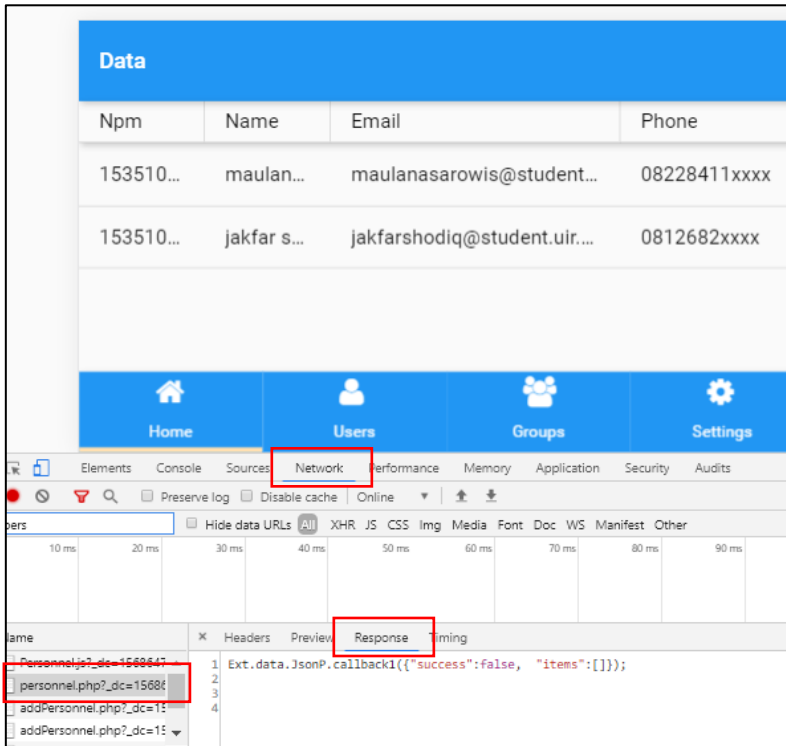
```



Perhatikan kode di atas, data dari tabel personnel dimasukkan dalam array \$output yang kemudian di konversi ke dalam format json dengan menggunakan fungsi `json_encode($output)`.

Refresh aplikasi di browser modern seperti Google Chrome, klik bagian Network, klik personnel.php dan lihat Response yang dihasilkan.





Gambar 34. View Respons

CRUD

CRUD adalah singkatan dari *Create Read Update Delete*, yang sering digunakan pada aplikasi-aplikasi pengolahan data yang kebanyakan menggunakan fungsi CRUD didalamnya . Fungsi ini digunakan untuk menambahkan



data, menghapus data, serta mengupdate data.

Sebelumnya, store Personnel masih menggunakan fungsi *read* pada atribut *api proxy*. Pada bagian ini, kita akan menambahkan fungsi *create*, *update*, dan *destroy*, dan menambahkan skrip PHP yang berada di http://localhost/MyApp_php/ yang berisi sintaks atau *query database* untuk *insert*, *update*, dan *delete*. Untuk menambahkan fungsi tersebut, kita menambahkan atribut api – create, update, dan delete. Buat file `app/store/Personnel.js` seperti berikut:



```

Ext.define('MyApp.store.Personnel', {
    extend: 'Ext.data.Store',
    alias: 'store.personnel',
    storeId: 'personnel',
    autoLoad: true,
    autoSync: true,
    fields: [
        'user_id', 'name', 'email', 'phone'
    ],
    proxy: {
        type: 'jsonp',
        api: {
            read: "http://localhost/MyApp_php/personnel.php",
            create : "http://localhost/MyApp_php/addPersonnel.php",
            update : "http://localhost/MyApp_php/updatePersonnel.php",
            destroy : http://localhost/MyApp\_php/removePersonnel.php
        }
    },
    reader: {
        type: 'json',
        rootProperty: 'items',
        messageProperty: 'error'
    }
});

```

Berikut detail skrip addPersonnel.php, updatePersonnel.php, removePersonnel.php dan auth.php tersebut:



- **addPersonnel.php**

```
<?php
include 'auth.php';
$callback = $_REQUEST['callback'];
$records = json_decode($_REQUEST['records']);

$user_id = $records->{"user_id"};
$name = $records->{"name"};
$email = $records->{"email"};
$phone = $records->{"phone"};

// Create the output object.
$output = array();
$success = 'false';
$query="insert into personnel (user_id,name,email,phone) values
('$user_id','$name','$email','$phone)";
if ($conn->query($query) === TRUE) {
    $success = 'true';
}
else{
    $success = 'false';
    $error = $conn->error;
}

//start output
if ($callback) {
    header('Content-Type: text/javascript');
    echo $callback . '({"success":' . $success . ', "items":' .
json_encode($output) . '})';
} else {
    header('Content-Type: application/x-json');
    echo json_encode($output);
}
$conn->close();
?>
```



- **updatePersonnel.php**

```
<?php
include 'auth.php';
$callback = $_REQUEST['callback'];
$records = json_decode($_REQUEST['records']);
$user_id= $records->{"user_id"};
$name = $records->{"name"};
$email = $records->{"email"};
$phone = $records->{"phone"};

$success = 'false';
$error = 'no error';
// Create the output object.
$output = array();

$query="update personnel set
name='$name',email='$email',phone='$phone' where
user_id=$user_id";

if ($conn->query($query) === TRUE) {
    $success = 'true';
}
else{
    $error = $conn->error;
}

//start output
if ($callback) {
    header('Content-Type: text/javascript');
    echo $callback . '({ "success":'. $success.',
"message":'. $error.'});';
} else {
    header('Content-Type: application/x-json');
    echo json_encode($output);
}
$conn->close();
?>
```



- **removePersonnel.php**

```
<?php
include 'auth.php';
$callback = $_REQUEST['callback'];
$records = json_decode($_REQUEST['records']);
$user_id = $records->{"user_id"};

$success = 'false';
$error = 'no error';
// Create the output object.
$output = array();
$query="delete from personnel where user_id=$user_id";
if ($conn->query($query) === TRUE) {
    $success = 'true';
}
else{
    $error = $conn->error;
}

//start output
if ($callback) {
    header('Content-Type: text/javascript');
    echo $callback . '({"success":'.$success.',
"message":'.$error.'});';
} else {
    header('Content-Type: application/x-json');
    echo json_encode($output);
}
$conn->close();
?>
```



- **auth.php**

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myapp";

date_default_timezone_set('Asia/Jakarta');
$time = date('Y-m-d H:i:s');
$conn = new mysqli($servername, $username, $password,
$dbname);
/* check connection */
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

Semua file php diatas disimpan pada xampp/htdocs di dalam folder yang sama.

Untuk input data, kita memerlukan register form sebagai media untuk user menginputkan data. Sintaks file app/view/main/RegisterForm.js form tersebut ialah sebagai berikut:



```

Ext.define('MyApp.view.main.RegisterForm', {
    extend: 'Ext.form.Panel',
    xtype: 'form-register',
    //frame: true,
    title: 'Register',
    controller: 'form-register',
    requires: [
        'Ext.form.FieldSet',
        'Ext.field.Email',
        'MyApp.view.main.RegisterFormController'
    ],
    shadow: true,
    cls: 'demo-solid-background',
    id: 'basicform',
    items: [
        {
            xtype: 'fieldset',
            id: 'fieldset1',
            title: 'Input Data',
            instructions: 'Silakan masukkan informasi di atas',
            defaults: {
                labelWidth: '35%'
            },
            items: [
                {
                    xtype: 'textfield',
                    //tambahan
                    id: 'user_id',

                    label: 'NPM',
                    placeholder: '153510xxx',
                    required: true,
                    clearIcon: true
                },
                {
                    xtype: 'textfield',
                    id: 'name',
                    label: 'Name',
                    placeholder: 'Nama Lengkap',

```



```

        required: true,
        clearIcon: true
    }, {
        xtype: 'emailfield',
        id: 'email',
        label: 'Email',
        placeholder: 'me@sencha.com',
        clearIcon: true
    },
    {
        xtype: 'textfield',
        id: 'phone',
        label: 'Phone',
        placeholder: '08228411xxxx',
        required: true,
        clearIcon: true
    },
]
},
{
    xtype: 'container',
    defaults: {
        xtype: 'button',
        style: 'margin: 1em',
        flex: 1
    },
    layout: {
        type: 'hbox'
    },
    items: [
        {
            text: 'Simpan',
            ui: 'action',
            formBind: true,
            handler: 'onRegisterClick',
        },
    ],
}

```



```
        {
            text: 'Reset',
            ui: 'action',
            handler: function(){
                Ext.getCmp('basicform').reset();
            }
        }
    ]
}
]
});
```

Untuk menandai data berhasil disimpan ke *database*, diperlukan fungsi pada *View Controller* untuk memberikan informasi bahwa data berhasil tersimpan pada file `app/view/main/RegisterFormController.js` seperti berikut:



```

Ext.define('MyApp.view.main.RegisterFormController', {
    extend: 'Ext.app.ViewController',

    alias: 'controller.form-register',

    onRegisterClick: function () {

        var user_id, name, email, phone, store;
        user_id = Ext.getCmp('user_id').getValue();
        name = Ext.getCmp('name').getValue();
        email = Ext.getCmp('email').getValue();
        phone = Ext.getCmp('phone').getValue();

        store = Ext.getStore('personnel');
        store.insert(0, [{ user_id: user_id, name: name, email:
email, phone}]);
        store.sync();

        Ext.getCmp('basicform').reset();

        Ext.Msg.alert('Berhasil', 'Pengguna baru telah ditambahkan',
Ext.emptyFn);

        console.log('store: ', store);
    }
});

```



Untuk *update* dan *delete* data, kita akan mencoba merubah data dari table dan menghapus data menggunakan tombol pada kolom hapus pada setiap baris. Untuk itu ditambahkan *plugin* dengan atribut *ptype* : ‘*cellediting*’ untuk *update* data dan untuk menghapus data diperlukan fungsi dengan action *remove*. Skrip file `app/view/main/List.js` ialah seperti berikut:



```
Ext.define('MyApp.view.main.List', {
    extend: 'Ext.grid.Grid',
    xtype: 'mainlist',

    requires: [
        'MyApp.store.Personnel',
    ],

    title: 'Data',

    //
    plugins: {
        type: 'grideditable',
        triggerEvent: 'doubletap',
        enableDeleteButton: true,
        formConfig: null,

        defaultFormConfig: {
            xtype: 'formpanel',
            scrollable: true,
            items: {
                xtype: 'fieldset'
            }
        }
    },
},
```



```

toolbarConfig: {
  xtype: 'titlebar',
  docked: 'top',
  items: [{
    xtype: 'button',
    ui: 'decline',
    text: 'Batal',
    align: 'left',
    action: 'cancel'
  }, {
    xtype: 'button',
    ui: 'confirm',
    text: 'Simpan',
    align: 'right',
    action: 'submit'
  }]
},
},
store: {
  type: 'personnel'
},

columns: [
  //{xtype: 'rownumberer'},
  { text: 'Npm', dataIndex: 'user_id', width: 100, editable:
false},
  { text: 'Name', dataIndex: 'name', width: 100, editable:
true},
  { text: 'Email', dataIndex: 'email', width: 230, editable: true},
  { text: 'Phone', dataIndex: 'phone', width: 150, editable:
true},
],
listeners: {

}
});

```



Untuk *event* pada setiap tombol, diperlukan file *control* `app/view/main/MainController.js` sebagai berikut:

```
Ext.define('MyApp.view.main.MainController', {
    extend: 'Ext.app.ViewController',

    alias: 'controller.main',

    onItemSelected: function (sender, record) {
        Ext.Msg.confirm('Confirm', 'Are you sure?', 'onConfirm', this);
    },

    onConfirm: function (choice) {
        if (choice === 'yes') {
        }
    }
});
```

Untuk tampilan berisi semua aspek visual aplikasi ini, diperlukan sintaks file `app/view/main/Main.js` sebagai berikut:

```
Ext.define('MyApp.view.main.Main', {
    extend: 'Ext.tab.Panel',
    xtype: 'app-main',

    requires: [
        'Ext.MessageBox',

        'MyApp.view.main.MainController',
        'MyApp.view.main.MainModel',
        'MyApp.view.main.List',
        'MyApp.view.main.RegisterForm'
    ],
```



```

controller: 'main',
viewModel: 'main',

defaults: {
  tab: {
    iconAlign: 'top'
  },
  styleHtmlContent: true
},

tabBarPosition: 'bottom',

items: [
  {
    title: 'Home',
    iconCls: 'x-fa fa-home',
    layout: 'fit',
    // The following grid shares a store with the classic version's
grid as well!
    items: [{
      xtype: 'mainlist'
    }]
  }, {
    title: 'Users',
    iconCls: 'x-fa fa-user',
    layout: 'fit',
    // The following grid shares a store with the classic version's
grid as well!
    items: [{
      xtype: 'form-register'
    }]
  }, {
    title: 'Groups',
    iconCls: 'x-fa fa-users',
    bind: {
      html: '{loremIpsum}'
    }
  }
],

```

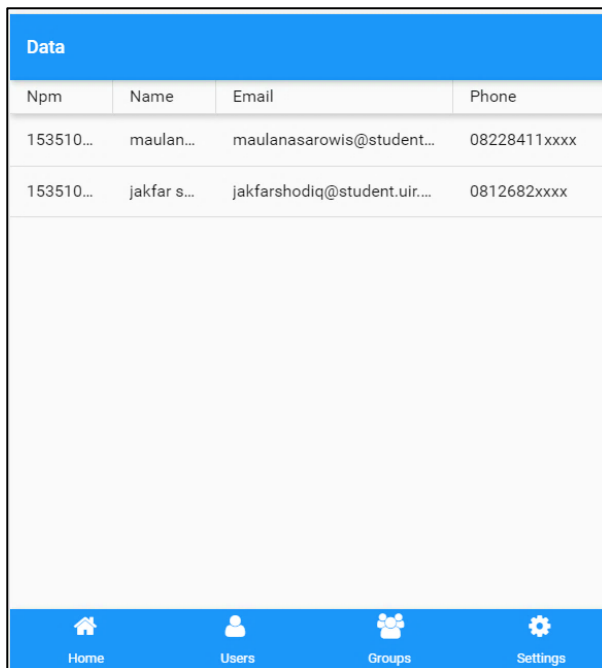


```
{
  title: 'Settings',
  iconCls: 'x-fa fa-cog',
  bind: {
    html: '{loremIpsum}'
  }
}
]
});
```

Refresh aplikasi di browser modern seperti Google Chrome.

Tampilan form aplikasi:

| Data | | | |
|-----------|-------------|-----------------------------|--------------|
| Npm | Name | Email | Phone |
| 153510... | maulan... | maulanasarowis@student... | 08228411xxxx |
| 153510... | jakfar s... | jakfarshodiq@student.uir... | 0812682xxxx |



Gambar 35. Tampilan Aplikasi



Tampilan menginputkan data ke form, isi semua kolom yang tersedia. Lalu klik tombol simpan, seperti berikut:

The image shows a mobile application interface for a registration form. At the top, there is a blue header with the word "Register" in white. Below the header, the form is titled "Input Data" and contains four input fields, each with a clear (X) button on the right:

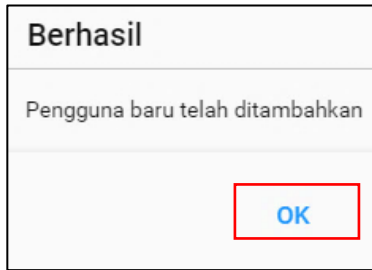
- NPM*: 153510357
- Name*: jakfar shodiq
- Email: jakfarshodiq@student.uir.ac.id
- Phone*: 0812682xxxx

Below the input fields, there is a small text prompt: "Silakan masukkan informasi di atas". Underneath this prompt are two blue buttons: "SIMPAN" (Save) and "RESET". The "SIMPAN" button is highlighted with a red rectangular border. At the bottom of the screen, there is a blue navigation bar with four icons and labels: "Home" (house icon), "Users" (person icon), "Groups" (group of people icon), and "Settings" (gear icon).

Gambar 36. Tambah Data

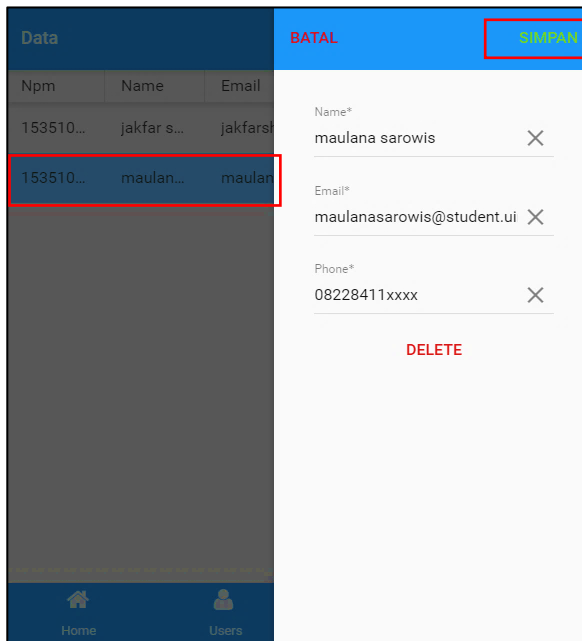


Berikut tampilan jika data berhasil disimpan:



Gambar 37. Pesan Konfirmasi

Berikut tampilan untuk mengubah data, klik dua kali pada cell yang akan diubah. lalu ketik data yang baru:



Gambar 38. Update Data



Berikut tampilan jika data berubah:

| Data | | | |
|-----------|-------------|-----------------------------|--------------|
| Npm | Name | Email | Phone |
| 153510... | jakfar s... | jakfarshodiq@student.uir... | 0812682xxxx |
| 153510... | maulana | maulana@student.uir.ac.id | 08228411xxxx |

Gambar 39. Data Setelah di Update

Tampilan menghapus data, klik delete dibawah untuk menghapus baris yang diinginkan:

| Data | | | BATAL | SIMPAN |
|-----------|-------------|----------|---------------------------|--------|
| Npm | Name | Email | Name* | |
| 153510... | jakfar s... | jakfarst | maulana | ✕ |
| 153510... | maulana | maular | Email* | |
| | | | maulana@student.uir.ac.id | ✕ |
| | | | Phone* | |
| | | | 08228411xxxx | ✕ |
| | | | DELETE | |

Gambar 40. Delete Data



Berikut tampilan setelah data terhapus:

| Data | | | |
|-----------|-------------|-----------------------------|-------------|
| Npm | Name | Email | Phone |
| 153510... | jakfar s... | jakfarshodiq@student.uir... | 0812682xxxx |

Gambar 41. Data Setelah di hapus

Latihan

1. Buat aplikasi sederhana dengan menggunakan *database* dengan menerapkan CRUD seperti di atas.



GLOSARIUM

| | |
|----------------------------------|--|
| Android | Sistem operasi mobile yang dikembangkan oleh Google. Sistem ini dirancang khusus untuk perangkat menggunakan layar sentuh seperti smartphone dan tablets |
| Aplikasi Mobile | Perangkat lunak yang dirancang khusus digunakan pada perangkat mobile seperti smartphone, tablet atau jam tangan |
| Aplikasi Web | Program komputer berbasis client-server dimana client menjalankan web browser. |
| Carousel | Cara yang bagus untuk memungkinkan pengguna menggesek (swipe) melalui beberapa halaman layar penuh |
| Cascading Style Sheets (CSS) | Bahasa pemrograman khusus untuk menjelaskan presentasi dari sebuah dokumen yang ditulis dalam bahasa markup seperti HTML |
| Command Line/Command Prompt | Sebuah tampilan tatap muka yang dinavigasikan dengan mengetikkan perintah pada prompts |
| Container | Jenis komponen khusus yang dapat mengandung komponen lain |
| Controller | pengendali keseluruhan aplikasi di sencha |
| Cordova | Distribusi dari Apache Cordova yang open-source |
| Create Read Update Delete (CRUD) | Fungsi yang digunakan untuk menambahkan data, menghapus data, serta mengupdate data. |
| Environment Variable | Sebuah nilai dinamik yang akan mempengaruhi cara proses kerja komputer |
| Ext JS | Framework aplikasi JavaScript untuk membangun platform yang interaktif antar aplikasi web menggunakan teknik seperti Ajax, DHTML dan DOM |



| | |
|-----------------------------------|--|
| Ext JS Kitchen Sink | Kumpulan tools yang kodifikasi dapat digunakan kembali |
| Ext JS Themes | Berisikan tema dari Ext JS |
| Folder | Sebuah kontainer atau wadah virtual dalam sistem file digital dengan mengelompokkan file dan mengorganisir folder lain |
| Forms | Panel dasar dengan kemampuan penanganan formulir seperti validasi data dan mendapatkan seluruh nilai dari komponen isian form tersebut |
| Grids | Komponen yang memaparkan data dalam bentuk tabel dengan memungkinkan pemaparan beberapa kolom |
| HTML5 | Sebuah software yang mendefinisikan properti dari isi halaman web dengan mengimplementasikan markup |
| IDE | Aplikasi perangkat lunak untuk pembangunan perangkat lunak |
| IOS | Sistem operasi perangkat bergerak yang dibangun dan didistribusikan oleh Apple. Sistem operasi ini dikembangkan untuk mendukung perangkat Apple lainnya seperti iPad. Apple tidak melisensikan iOS untuk diinstal di perangkat keras non-Apple. |
| Java | Bahasa pemrograman berbasis objek yang banyak dijalankan pada berbagai macam sistem operasi seperti Android |
| Javascript Object Notation (JSON) | Merupakan metadata yang menggunakan format yang mudah dibaca untuk melakukan pertukaran data objek yang terdiri dari pasangan atribut-nilai dan tipe data array |
| JQTouch | Sebuah plugin atau JQuery open-source dengan animasi asli, navigasi otomatis, dan tema untuk browser Webkit seperti iPhone sehingga memungkinkan programmer mengembangkan aplikasi mobile dengan tampilan asli untuk perangkat menggunakan HTML, CSS, dan JavaScript |



| | |
|----------------------|---|
| JSONP | Merupakan mekanisme untuk melakukan pertukaran data antar domain |
| Layout | Salah satu bagian Ext JS yang berperan penting dalam menangani ukuran dan posisi setiap komponen dalam aplikasi yang dibangun |
| Library | Kumpulan sumber non-volatile yang digunakan oleh program komputer untuk pengembangan perangkat lunak diantaranya termasuk data konfigurasi, dokumentasi, data bantuan, templat pesan, kode pra-tertulis dan subrutin, kelas, nilai atau spesifikasi tipe. |
| Lifecycle Management | Mencakup rancangan (<i>scaffolding</i>), minifikasi koding, pembuatan aplikasi dan lain sebagainya. |
| List | Komponen ini memaparkan data dalam bentuk daftar |
| Listeners Select | Menghubungkan data yang dipilih (dengan tap/sentuhan) dengan fungsi 'onItemSelected' yang dikelola oleh <i>ViewController</i> |
| Localhost | Sebuah hostname yang menandakan bahwa komputer ini digunakan untuk mengakses jaringan lokal/setempat |
| Metadata | Informasi data yang menyediakan informasi tentang data lainnya |
| Microloader | Memuat kode-kode aplikasi yang dijelaskan dalam file <i>app.json</i> |
| Model | Merepresentasikan bentuk object yang berfungsi sebagai penghubung ke database dan memberitahukan Controller untuk merubah View |
| Output Build | Hasil yang dibangun dengan menggunakan aplikasi sencha |
| Path | Bentuk umum dari nama file atau direktori yang menspesifikasikan sebuah lokasi unik dalam sebuah sistem file |
| Phonegap | Versi komersial Cordova dari Adobe beserta ekosistem yang terkait. Banyak alat dan kerangka kerja lain juga |



dibangun di atas Cordova, termasuk Ionic

| | |
|---|--|
| Platforms | Dasar perangkat keras (computer) dan perangkat luna (sistem operasi) tempat aplikasi perangkat keras dijalankan |
| Plugin | Sebuah komponen perangkat lunak yang menambahkan fitur spesifik ke program komputer yang telah tersedia |
| Proxy | Memungkinkan transfer data antara penyedia data dan aplikasi Anda |
| Sencha Cmd | Seperangkat <i>command line</i> lintas platform atau dikenal dengan <i>library</i> yang menyediakan fitur lengkap mencakup <i>lifecycle management</i> mulai dari pembuatan proyek baru sampai dengan menjalankan (<i>deploy</i>) aplikasi yang kemudian dikemas menjadi suatu produk. |
| Sencha Web Application Manager Server Store | Memungkinkan akses aman ke server otentikasi dan otorisasi perusahaan Data di sisi klien (merujuk ke definisi data di kelas Model). Store menyediakan fungsi untuk menyortir, memfilter dan memuat data yang ada di dalamnya |
| Sub-Folder | Folder di dalam folder |
| Sublime | perangkat lunak yang berfungsi sebagai editor penulisan bahasa pemrograman lintas platform dengan antarmuka pemrograman aplikasi Python |
| System Environment Variables Tabpanel | Menu untuk mendaftarkan lokasi aplikasi seperti sencha cmd sehingga dapat diakses di <i>command prompt</i> Menu pada Sencha yang memungkinkan pengguna untuk beralih di antara beberapa halaman dengan tampilan layar penuh. |
| Tizen | Sistem operasi seluler berbasis Linux dikembangkan dan digunakan terutama oleh Samsung Electronics. |



| | |
|-----------------|---|
| Tuning Tools | Alat untuk menyelesikan kode untuk menyetel apa yang dimasukkan dalam aplikasi akhir, memilih kode umum lintas halaman dan kode partisi menggunakan operasi tingkat tinggi untuk membangun aplikasi yang diinginkan |
| User Interface | Tampilan tatap muka pada perangkat lunak di komputer yang dilihat oleh pengguna untuk kemudian mengakses konten yang ada di dalamnya |
| View | Merupakan visualisasi yang dilihat oleh pengguna yang berperan sebagai tampilan muka (interface) dari aplikasi dan memberitahu Controller merubah Model sesuai input dari user |
| View Controller | Memberikan informasi bahwa data berhasil tersimpan pada file |
| View Model | Mendefinisikan data yang dibuat di Main Model |
| Viewport | Menghubungkan <i>View</i> ke badan dokumen html |



INDEKS

A

Android, 9, 65, 91, 92
Aplikasi mobile, 91
Aplikasi web, 91

C

Carousel, 6, 7, 52, 53, 54, 55, 56,
63, 91
Command line, 91
Command prompt, 91
Container, 6, 49, 57, 91
Controller, 6, 7, 11, 27, 33, 34,
43, 79, 91, 93, 94
Cordova, 12, 91, 93
CRUD, 6, 70, 91
CSS, 91, 92

E

Environment Variable, 18, 91
Ext JS, 7, 9, 10, 11, 13, 22, 24,
25, 26, 36, 37, 49, 91, 92

F

Folder, 7, 26, 35, 36, 92, 94
Forms, 6, 7, 57, 64, 92

G

Grids, 6, 55, 92

H

HTML5, 9, 92

I

IDE, 32, 92
iOS, 9, 92

J

Java, 7, 13, 92
jQTouch, 9, 92
JSONP, 65, 92

K

Kitchen Sink, 7, 24, 25, 91

L

layout, 49, 92
library, 5, 10, 30, 93, 94
lifecycle management, 10, 93, 94
List, 6, 7, 42, 46, 55, 56, 81, 93
listeners select, 42, 93
localhost, 11, 23, 24, 31, 65, 71,
93

M

metadata, 36, 65, 92, 93
Microloader, 36, 93
Model, 6, 7, 11, 27, 33, 34, 45,
57, 93, 94



O

output build, 31, 93

P

Path, 19, 93
PhoneGap, 12, 93
platforms, 10, 93
Plugin, 39, 93
Proxy, 46, 93

S

Sencha Cmd, 7, 10, 11, 13, 14,
15, 16, 21, 22, 30, 31, 36, 37,
94
Sencha Web Application
Manager server, 11, 94
Store, 6, 27, 45, 94
sub-folder, 22, 27, 94
Sublime, 6, 32, 35, 94

System Environment Variables,
18, 94

T

TabPanel, 38, 94
Tizen, 10, 94
Tuning Tools, 12, 94

U

User Interface, 94

V

View, 6, 7, 8, 11, 27, 33, 34, 39,
41, 42, 43, 45, 46, 70, 79, 93,
94, 95
View Controller, 94
View Model, 94
Viewport, 38, 39, 95



BIOGRAFI PENULIS

| | |
|---|--|
|  | <p>Dr. Arbi Haza Nasution, B.IT (Hons), M.IT merupakan dosen tetap pada Jurusan Teknik Informatika, Universitas Islam Riau. Pendidikan S1 ditempuh di Jurusan Ilmu Komputer, Universiti Kebangsaan Malaysia (2010). Pendidikan Magister di Management Sistem Informasi pada universitas yang sama (2012). Beliau merupakan penerima Beasiswa President Republik Indonesia (BPRI) yang dikelola oleh Lembaga Pengelolaan Dana Pendidikan (LPDP) Kementerian Keuangan Republik Indonesia untuk program doctoral yang diselesaikan pada kurun waktu tiga setengah tahun di Kyoto University (2018). Sejumlah prestasi nasional dan internasional pernah beliau raih. Di samping mengajar, beliau juga aktif sebagai peneliti dengan spesifikasi bidang <i>computational linguistics</i>, <i>natural language processing</i> dan <i>machine learning</i>.</p> |
|  | <p>Winda Monika, S.Pd., M.LIS merupakan pustakawan di Universitas Islam Riau. Pendidikan S1 ditempuh di Jurusan Perpustakaan dan Informasi, Universitas Pendidikan Indonesia (2013). Beliau merupakan penerima Beasiswa Pendidikan Indonesia (BPI) yang dikelola oleh Lembaga Pengelolaan Dana Pendidikan (LPDP) Kementerian Keuangan Republik Indonesia untuk program magister di Graduate School of Library, Information and Media Studies, University of Tsukuba (2018). Beliau aktif sebagai peneliti dengan spesifikasi bidang <i>Big Data</i>, <i>Metadata</i>, <i>Semantic Web</i> dan <i>Linked Open Data</i>.</p> |
|  | <p>Salhazan Nasution, S.Kom., M.IT memperoleh gelar Sarjana Komputer (S.Kom) dari Teknik Informatika Universitas Islam Indonesia pada tahun 2008, memperoleh gelar Master of Information Technology (M.IT) dari Universiti Kebangsaan Malaysia pada tahun 2015. Merupakan dosen tetap di Prodi Teknik Informatika Universitas Riau sejak tahun 2014. Saat ini bidang penelitian yang dilakukan meliputi <i>Management Information System</i>, <i>Mobile Application</i>, <i>Internet of Things</i> (IoT) dan Sistem Informasi Geografis (SIG).</p> |

