M.Bahrul Ulum, S.Kom, M.Kom





Pemrograman Mobile

Menu

**MENU**

Menu adalah komponen antarmuka pengguna yang lazim dalam banyak tipe aplikasi. Untuk menyediakan pengalaman pengguna yang sudah akrab dan konsisten, Anda harus menggunakan API [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=id) untuk menyajikan tindakan dan opsi lain dalam aktivitas kepada pengguna.

Mulai dengan Android 3.0 (API level 11), perangkat Android tidak perlu lagi menyediakan tombol *Menu*tersendiri. Dengan perubahan ini, aplikasi Android harus bermigrasi dari dependensi pada panel menu 6 item biasa, dan sebagai gantinya menyediakan bilah aplikasi untuk menyajikan berbagai tindakan pengguna yang lazim.

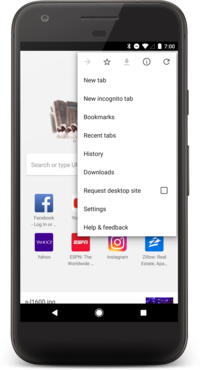
Walaupun desain dan pengalaman pengguna untuk sebagian item menu telah berubah, semantik untuk mendefinisikan serangkaian tindakan dan opsi masih berdasarkan pada [Menu](https://developer.android.com/reference/android/view/Menu.html?hl=id) API. Panduan ini menampilkan cara membuat tiga tipe dasar penyajian menu atau aksi pada semua versi Android :

**MENU OPSI**

[Menu Opsi](https://developer.android.com/guide/topics/ui/menus?hl=id#options-menu) adalah kumpulan item menu utama untuk suatu aktivitas. Inilah tempat Anda harus menempatkan tindakan yang berdampak global pada aplikasi, seperti "Cari", "Tulis email", dan "Setelan".

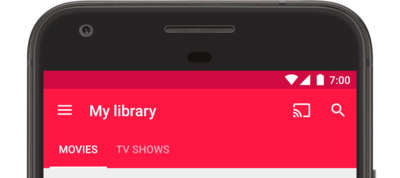
Tempat item dalam menu opsi muncul di layar bergantung pada versi aplikasi yang Anda kembangkan:

* Jika Anda mengembangkan aplikasi untuk **Android 2.3.x (API level 10) atau yang lebih rendah**, materi menu opsi akan muncul di bagian bawah layar bila pengguna menekan tombol *Menu*, seperti yang ditampilkan dalam gambar 1. Bila dibuka, bagian yang terlihat pertama adalah menu ikon, yang menampung hingga enam item menu. Jika menu Anda menyertakan lebih dari enam item, Android akan meletakkan item keenam dan sisanya ke dalam menu luapan, yang bisa dibuka pengguna dengan memilih *More*.



**Gambar 1.** Menu opsi di Browser, pada Android 2.3.

* Jika Anda mengembangkan aplikasi untuk **Android 3.0 (API level 11) dan yang lebih tinggi**, item menu opsi tersedia dalam bilah aplikasi. Secara default, sistem meletakkan semua item dalam aksi selebihnya, yang bisa ditampilkan pengguna dengan ikon aksi selebihnya di sisi kanan bilah aplikasi (atau dengan menekan tombol *Menu* perangkat, jika tersedia). Untuk mengaktifkan akses cepat ke tindakan penting, Anda bisa mempromosikan beberapa item agar muncul pada bilah aplikasi dengan menambahkan android:showAsAction="ifRoom" ke elemen-elemen <item> yang bersangkutan (lihat gambar 2).



**Gambar 2.** Bilah aplikasi, yang menampilkan tab-tab navigasi dan item aksi kamera (plus tombol aksi selebihnya).

**MENU KONTEKS DAN MODE AKSI KONTEKSTUAL**

Menu konteks adalah [menu mengambang](https://developer.android.com/guide/topics/ui/menus?hl=id#FloatingContextMenu) yang muncul saat pengguna mengeklik lama pada suatu elemen. Menu ini menyediakan tindakan yang memengaruhi materi atau bingkai konteks yang dipilih.

[Mode aksi kontekstual](https://developer.android.com/guide/topics/ui/menus?hl=id#CAB) menampilkan item aksi yang memengaruhi materi yang dipilih dalam bilah di bagian atas layar dan memungkinkan pengguna memilih beberapa item sekaligus.

Membuat Menu Kontekstual



**Gambar 3.** Tangkapan layar menu konteks mengambang (kiri) dan bilah aksi kontekstual (kanan).

Menu kontekstual menawarkan tindakan yang memengaruhi item atau bingkai konteks tertentu dalam UI. Anda bisa menyediakan menu konteks untuk tampilan apa saja, namun menu ini paling sering digunakan untuk item dalam [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=id), [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=id), atau kumpulan tampilan lainnya yang bisa digunakan pengguna untuk melakukan tindakan langsung pada setiap item.

Ada dua cara menyediakan tindakan kontekstual:

* Dalam [menu konteks mengambang](https://developer.android.com/guide/topics/ui/menus?hl=id#FloatingContextMenu). Menu muncul sebagai daftar item menu mengambang (serupa dengan dialog) bila pengguna mengeklik lama (menekan dan menahan) pada tampilan yang mendeklarasikan dukungan bagi menu konteks. Pengguna bisa melakukan aksi kontekstual pada satu item untuk setiap kalinya.
* Dalam [mode aksi kontekstual](https://developer.android.com/guide/topics/ui/menus?hl=id#CAB). Mode ini adalah implementasi sistem [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=id) yang menampilkan bilah aksi kontekstual di bagian atas layar dengan item aksi yang memengaruhi item(-item) yang dipilih. Bila mode ini aktif, pengguna bisa melakukan aksi pada beberapa item sekaligus (jika aplikasi Anda mengizinkannya).

**Catatan:** Mode aksi kontekstual tersedia pada Android 3.0 (API level 11) dan yang lebih tinggi dan merupakan teknik yang lebih disukai untuk menampilkan aksi kontekstual bila tersedia. Jika aplikasi Anda mendukung versi yang lebih rendah daripada 3.0, maka Anda harus mundur ke menu konteks mengambang pada perangkat-perangkat itu.

**Membuat menu konteks mengambang**

Untuk menyediakan menu konteks mengambang:

1. Daftarkan [View](https://developer.android.com/reference/android/view/View.html?hl=id) ke menu konteks yang harus dikaitkan dengan memanggil [registerForContextMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=id#registerForContextMenu(android.view.View))dan teruskan [View](https://developer.android.com/reference/android/view/View.html?hl=id) ke menu itu.

Jika aktivitas Anda menggunakan [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=id) atau [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=id) dan Anda ingin setiap item untuk menyediakan menu konteks yang sama, daftarkan semua item ke menu konteks dengan meneruskan [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=id) atau [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=id) ke [registerForContextMenu()](https://developer.android.com/reference/android/app/Activity.html?hl=id#registerForContextMenu(android.view.View)).

1. Implementasikan metode [onCreateContextMenu()](https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener.html?hl=id#onCreateContextMenu(android.view.ContextMenu,%20android.view.View,%20android.view.ContextMenu.ContextMenuInfo)) dalam [Activity](https://developer.android.com/reference/android/app/Activity.html?hl=id) atau [Fragment](https://developer.android.com/reference/android/app/Fragment.html?hl=id).

Bila tampilan yang terdaftar menerima kejadian klik-lama, sistem akan memanggil metode [onCreateContextMenu()](https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener.html?hl=id#onCreateContextMenu(android.view.ContextMenu,%20android.view.View,%20android.view.ContextMenu.ContextMenuInfo)) Anda. Inilah tempat Anda mendefinisikan item menu, biasanya dengan memekarkan sumber daya menu. Misalnya :

@Override  
public void onCreateContextMenu(ContextMenu menu, View v,  
                                ContextMenuInfo menuInfo) {  
    super.onCreateContextMenu(menu, v, menuInfo);  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.context\_menu, menu);  
}

[Menu Inflater](https://developer.android.com/reference/android/view/MenuInflater.html?hl=id) memungkinkan Anda untuk memekarkan menu konteks [sumber daya menu](https://developer.android.com/guide/topics/resources/menu-resource.html?hl=id). Parameter metode callback menyertakan [View](https://developer.android.com/reference/android/view/View.html?hl=id) yang dipilih pengguna dan objek [ContextMenu.ContextMenuInfo](https://developer.android.com/reference/android/view/ContextMenu.ContextMenuInfo.html?hl=id)yang menyediakan informasi tambahan tentang item yang dipilih. Jika aktivitas Anda memiliki beberapa tampilan yang masing-masingnya menyediakan menu konteks berbeda, Anda bisa menggunakan parameter ini untuk menentukan menu konteks yang harus dimekarkan.

1. Implementasikan [onContextItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=id#onContextItemSelected(android.view.MenuItem)).

Bila pengguna memilih item menu, sistem akan memanggil metode ini sehingga Anda bisa melakukan aksi yang sesuai. Misalnya:

@Override  
public boolean onContextItemSelected(MenuItem item) {  
    AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();  
    switch (item.getItemId()) {  
        case R.id.edit:  
            editNote(info.id);  
            return true;  
        case R.id.delete:  
            deleteNote(info.id);  
            return true;  
        default:  
            return super.onContextItemSelected(item);  
    }  
}

Metode [getItemId()](https://developer.android.com/reference/android/view/MenuItem.html?hl=id#getItemId()) melakukan kueri ID untuk item menu yang dipilih, yang harus Anda tetapkan ke setiap item menu dalam XML dengan menggunakan atribut android:id, seperti yang ditampilkan di bagian tentang [Mendefinisikan Menu dalam XML](https://developer.android.com/guide/topics/ui/menus?hl=id#xml).

Bila Anda berhasil menangani sebuah item menu, kembalikan true. Jika tidak menangani item menu, Anda harus meneruskan item menu ke implementasi superkelas. Jika aktivitas Anda menyertakan fragmen, aktivitas akan menerima callback ini lebih dahulu. Dengan memanggil superkelas bila tidak ditangani, sistem meneruskan kejadian ke metode callback di setiap fragmen, satu per satu (sesuai dengan urutan penambahan fragmen) hingga true atau false dikembalikan. (Implementasi default untuk [Activity](https://developer.android.com/reference/android/app/Activity.html?hl=id) dan android.app.Fragment mengembalikan false, sehingga Anda harus selalu memanggil superkelas bila tidak ditangani.)

**Menggunakan mode aksi kontekstual**

Mode aksi kontekstual adalah implementasi sistem [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=id) yang memfokuskan interaksi pengguna pada upaya melakukan aksi kontekstual. Bila seorang pengguna mengaktifkan mode ini dengan memilih item, bilah aksi kontekstual akan muncul di bagian atas layar untuk menampilkan tindakan yang bisa dilakukan pengguna pada item yang dipilih saat ini. Selagi mode ini diaktifkan, pengguna bisa memilih beberapa item (jika Anda mengizinkan), membatalkan pilihan item, dan melanjutkan penelusuran dalam aktivitas (sebanyak yang ingin Anda izinkan). Mode aksi dinonaktifkan dan bilah aksi kontekstual menghilang bila pengguna membatalkan pilihan semua item, menekan tombol BACK, atau memilih aksi Done di sisi kiri bilah aksi.

**Catatan:** Bilah aksi kontekstual tidak harus terkait dengan bilah aplikasi. Keduanya beroperasi secara independen, walaupun bilah aksi kontekstual secara visual mengambil alih posisi bilah aplikasi.

Untuk tampilan yang menyediakan tindakan kontekstual, Anda biasanya harus memanggil mode aksi kontekstual pada salah satu dari dua kejadian (atau keduanya):

* Pengguna mengeklik-lama pada tampilan.
* Pengguna memilih kotak centang atau komponen UI yang serupa dalam tampilan.

Cara aplikasi memanggil mode aksi kontekstual dan mendefinisikan perilaku setiap aksi bergantung pada desain Anda. Pada dasarnya ada dua desain:

* Untuk tindakan kontekstual pada tampilan arbitrer dan tak didukung.
* Untuk aksi kontekstual batch atas kelompok item dalam  [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=id)  atau  [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=id)  (memungkinkan pengguna memilih beberapa item dan melakukan aksi pada semua item itu).

Bagian berikut ini menjelaskan penyiapan yang diperlukan untuk setiap skenario.

#### Mengaktifkan mode aksi kontekstual untuk tampilan individual

Jika Anda ingin memanggil mode aksi kontekstual hanya bila pengguna memilih tampilan tertentu, Anda harus:

1. Mengimplementasikan antarmuka [ActionMode.Callback](https://developer.android.com/reference/android/view/ActionMode.Callback.html?hl=id). Dalam metode callback-nya, Anda bisa menetapkan tindakan untuk bilah aksi kontekstual, merespons kejadian klik pada item aksi, dan menangani kejadian daur hidup lainnya untuk mode aksi itu.
2. Memanggil [startActionMode()](https://developer.android.com/reference/android/app/Activity.html?hl=id#startActionMode(android.view.ActionMode.Callback)) bila Anda ingin menampilkan action-bar (seperti saat pengguna mengeklik-lama pada tampilan).

Misalnya:

1. Implementasikan antarmuka [ActionMode.Callback](https://developer.android.com/reference/android/view/ActionMode.Callback.html?hl=id) :

private ActionMode.Callback mActionModeCallback = new ActionMode.Callback() {  
  
    // Called when the action mode is created; startActionMode() was called  
    @Override  
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {  
        // Inflate a menu resource providing context menu items  
        MenuInflater inflater = mode.getMenuInflater();  
        inflater.inflate(R.menu.context\_menu, menu);  
        return true;  
    }  
  
    // Called each time the action mode is shown. Always called after onCreateActionMode, but  
    // may be called multiple times if the mode is invalidated.  
    @Override  
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {  
        return false; // Return false if nothing is done  
    }  
  
    // Called when the user selects a contextual menu item  
    @Override  
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {  
        switch (item.getItemId()) {  
            case R.id.menu\_share:  
                shareCurrentItem();  
                mode.finish(); // Action picked, so close the CAB  
                return true;  
            default:  
                return false;  
        }  
    }  
  
    // Called when the user exits the action mode  
    @Override  
    public void onDestroyActionMode(ActionMode mode) {  
        mActionMode = null; }};

Perhatikan, kejadian callback ini hampir persis sama dengan callback untuk [menu opsi](https://developer.android.com/guide/topics/ui/menus?hl=id#options-menu), hanya saja setiap callback ini juga meneruskan objek [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=id) yang terkait dengan kejadian. Anda bisa menggunakan [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=id) API untuk membuat berbagai perubahan pada CAB, seperti merevisi judul dan subjudul dengan [setTitle()](https://developer.android.com/reference/android/view/ActionMode.html?hl=id#setTitle(int)) dan [setSubtitle()](https://developer.android.com/reference/android/view/ActionMode.html?hl=id#setSubtitle(int)) (berguna untuk menunjukkan jumlah item yang dipilih).

Juga perhatikan kumpulan contoh di atas menyetel variabel mActionMode ke nol bila mode aksi dimusnahkan. Dalam langkah berikutnya, Anda akan melihat cara variabel diinisialisasi dan kegunaan menyimpan variabel anggota dalam aktivitas atau fragmen.

1. Panggil [startActionMode()](https://developer.android.com/reference/android/app/Activity.html?hl=id#startActionMode(android.view.ActionMode.Callback)) untuk mengaktifkan mode aksi kontekstual bila sesuai, seperti saat merespons klik-lama pada [View](https://developer.android.com/reference/android/view/View.html?hl=id):

someView.setOnLongClickListener(new View.OnLongClickListener() {  
    // Called when the user long-clicks on someView  
    public boolean onLongClick(View view) {  
        if (mActionMode != null) {  
            return false;  
        }  
  
        // Start the CAB using the ActionMode.Callback defined above  
        mActionMode = getActivity().startActionMode(mActionModeCallback);  
        view.setSelected(true);  
        return true;  
    }  
});

Bila Anda memanggil [startActionMode()](https://developer.android.com/reference/android/app/Activity.html?hl=id#startActionMode(android.view.ActionMode.Callback)), sistem akan mengembalikan [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=id) yang dibuat. Dengan menyimpannya dalam variabel anggota, Anda bisa membuat perubahan ke bilah aksi kontekstual sebagai respons terhadap kejadian lainnya. Dalam contoh di atas, [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=id) digunakan untuk memastikan bahwa instance [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html?hl=id) tidak dibuat kembali jika sudah aktif, dengan memeriksa apakah anggota bernilai nol sebelum memulai mode aksi.

**Mengaktifkan tindakan kontekstual batch dalam ListView atau GridView**

Jika Anda memiliki sekumpulan item dalam [ListView](https://developer.android.com/reference/android/widget/ListView.html?hl=id) atau [GridView](https://developer.android.com/reference/android/widget/GridView.html?hl=id) (atau ekstensi   [AbsListView](https://developer.android.com/reference/android/widget/AbsListView.html?hl=id) lainnya) dan ingin memungkinkan pengguna melakukan tindakan batch, Anda harus:

* Mengimplementasikan antarmuka [AbsListView.MultiChoiceModeListener](https://developer.android.com/reference/android/widget/AbsListView.MultiChoiceModeListener.html?hl=id)   dan mengaturnya untuk grup tampilan dengan [setMultiChoiceModeListener()](https://developer.android.com/reference/android/widget/AbsListView.html?hl=id#setMultiChoiceModeListener(android.widget.AbsListView.MultiChoiceModeListener)). Dalam metode callback listener, Anda bisa menetapkan tindakan untuk bilah aksi kontekstual, merespons kejadian klik pada item aksi, dan menangani callback lainnya yang diwarisi dari antarmuka [ActionMode.Callback](https://developer.android.com/reference/android/view/ActionMode.Callback.html?hl=id).
* Panggil [setChoiceMode()](https://developer.android.com/reference/android/widget/AbsListView.html?hl=id#setChoiceMode(int)) bersama argumen [CHOICE\_MODE\_MULTIPLE\_MODAL](https://developer.android.com/reference/android/widget/AbsListView.html?hl=id#CHOICE_MODE_MULTIPLE_MODAL).

Misalnya:

ListView listView = getListView();  
listView.setChoiceMode(ListView.CHOICE\_MODE\_MULTIPLE\_MODAL);  
listView.setMultiChoiceModeListener(new MultiChoiceModeListener() {  
  
    @Override  
    public void onItemCheckedStateChanged(ActionMode mode, int position,  
                                          long id, boolean checked) {  
        // Here you can do something when items are selected/de-selected,  
        // such as update the title in the CAB  
    }  
  
    @Override  
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {  
        // Respond to clicks on the actions in the CAB  
        switch (item.getItemId()) {  
            case R.id.menu\_delete:  
                deleteSelectedItems();  
                mode.finish(); // Action picked, so close the CAB  
                return true;  
            default:  
                return false;  
        }  
    }  
  
    @Override  
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {  
        // Inflate the menu for the CAB  
        MenuInflater inflater = mode.getMenuInflater();  
        inflater.inflate(R.menu.context, menu);  
        return true;  
    }  
  
    @Override  
    public void onDestroyActionMode(ActionMode mode) {  
        // Here you can make any necessary updates to the activity when  
        // the CAB is removed. By default, selected items are deselected/unchecked.  
    }  
  
    @Override  
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {  
        // Here you can perform updates to the CAB due to  
        // an [**invalidate()**](https://developer.android.com/reference/android/view/ActionMode.html?hl=id#invalidate()) request  
        return false;  
    }  
});

Demikian saja. Kini bila pengguna memilih item dengan klik-lama, sistem akan memanggil metode [onCreateActionMode()](https://developer.android.com/reference/android/view/ActionMode.Callback.html?hl=id#onCreateActionMode(android.view.ActionMode,%20android.view.Menu)) dan menampilkan bilah aksi kontekstual bersama tindakan yang ditetapkan. Saat bilah aksi kontekstual terlihat, pengguna bisa memilih item tambahan.

Dalam beberapa kasus di mana tindakan kontekstual menyediakan item aksi umum, Anda mungkin ingin menambahkan kotak centang atau elemen UI serupa yang memungkinkan pengguna memilih item, karena pengguna mungkin tidak menemukan perilaku klik-lama. Bila pengguna memilih kotak centang itu, Anda bisa memanggil mode aksi kontekstual dengan menyetel item daftar yang bersangkutan ke status diberi tanda centang dengan [setItemChecked()](https://developer.android.com/reference/android/widget/AbsListView.html?hl=id#setItemChecked(int,%20boolean)).

**GRUP MENU**

**Grup Menu** adalah sekumpulan item menu yang sama-sama memiliki ciri (trait) tertentu. Dengan grup, Anda bisa:

* Menampilkan atau menyembunyikan semua item dengan [setGroupVisible()](https://developer.android.com/reference/android/view/Menu.html?hl=id#setGroupVisible(int,%20boolean))
* Mengaktifkan atau mennonaktifkan semua item dengan [setGroupEnabled()](https://developer.android.com/reference/android/view/Menu.html?hl=id#setGroupEnabled(int,%20boolean))
* Menetapkan apakah semua item bisa dicentang dengan [setGroupCheckable()](https://developer.android.com/reference/android/view/Menu.html?hl=id#setGroupCheckable(int,%20boolean,%20boolean))

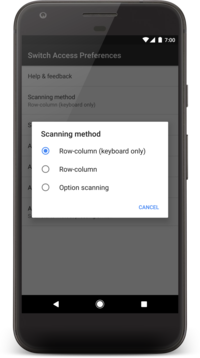
Anda bisa membuat kelompok dengan menyarangkan elemen-elemen <item> dalam elemen <group> di sumber daya menu atau dengan menetapkan ID kelompok bersama metode [add()](https://developer.android.com/reference/android/view/Menu.html?hl=id#add(int,%20int,%20int,%20int)).

Berikut ini adalah contoh sumber daya menu yang berisi sebuah grup:

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:id="@+id/menu\_save"  
          android:icon="@drawable/menu\_save"  
          android:title="@string/menu\_save" />  
    <!-- menu group -->  
    <group android:id="@+id/group\_delete">  
        <item android:id="@+id/menu\_archive"  
              android:title="@string/menu\_archive" />  
        <item android:id="@+id/menu\_delete"  
              android:title="@string/menu\_delete" />  
    </group>  
</menu>

Item yang berada dalam grup akan muncul pada level yang sama dengan item pertama—ketiga item dalam menu adalah bersaudara. Akan tetapi, Anda bisa memodifikasi ciri kedua item dalam grup dengan mengacu ID grup dan menggunakan metode yang tercantum di atas. Sistem juga tidak akan memisahkan item yang telah dikelompokkan. Misalnya, jika Anda mendeklarasikan android:showAsAction="ifRoom" untuk setiap item, item tersebut akan muncul dalam bilah aksi atau dalam kelebihan aksi.

### Menggunakan item menu yang bisa diberi tanda cek



**Gambar 5.** Tangkapan layar submenu bersama item yang bisa dicentang.

Menu bisa digunakan sebagai antarmuka untuk mengaktifkan dan menonaktifkan opsi, menggunakan kotak centang untuk opsi mandiri, atau tombol radio untuk grup opsi yang saling eksklusif. Gambar 5 menampilkan submenu dengan item yang bisa diberi tanda cek dengan tombol radio.

**Catatan:** Item menu dalam Icon Menu (dari menu opsi) tidak bisa menampilkan kotak centang atau tombol radio. Jika Anda memilih untuk membuat item dalam Icon Menu yang bisa diberi tanda cek, Anda harus menandai status diberi tanda cek secara manual dengan menukar ikon dan/atau teks tiap kali statusnya berubah.

Anda bisa mendefinisikan perilaku yang bisa dicentang untuk setiap item menu dengan menggunakan atribut android:checkable dalam elemen <item>, atau untuk seluruh kelompok dengan atribut android:checkableBehavior dalam elemen <group>. Misalnya , semua item dalam grup menu ini bisa diberi tanda cek dengan tombol radio:

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <group android:checkableBehavior="single">  
        <item android:id="@+id/red"  
              android:title="@string/red" />  
        <item android:id="@+id/blue"  
              android:title="@string/blue" />  
    </group>  
</menu>

Atribut android:checkableBehavior menerima:

Single Hanya satu item dari grup ini yang bisa diberi tanda cek (tombol radio)

All Semua item bisa diberi tanda cek (kotak centang)

None Tidak ada item yang bisa diberi tanda cek

Anda bisa menerapkan status dicentang default pada suatu item dengan menggunakan atribut android:checked dalam elemen <item> dan mengubahnya dalam kode dengan metode [setChecked()](https://developer.android.com/reference/android/view/MenuItem.html?hl=id#setChecked(boolean)).

Bila item yang bisa diberi tanda cek dipilih, sistem akan memanggil metode callback setiap item yang dipilih (seperti [onOptionsItemSelected()](https://developer.android.com/reference/android/app/Activity.html?hl=id#onOptionsItemSelected(android.view.MenuItem))). Di sinilah Anda harus mengatur status kotak centang itu, karena kotak centang atau tombol radio tidak mengubah statusnya secara otomatis. Anda bisa melakukan kueri status saat ini suatu item (seperti sebelum pengguna memilihnya) dengan [isChecked()](https://developer.android.com/reference/android/view/MenuItem.html?hl=id#isChecked()) kemudian mengatur status diberi tanda cek dengan [setChecked()](https://developer.android.com/reference/android/view/MenuItem.html?hl=id#setChecked(boolean)). Misalnya:

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.vibrate:  
        case R.id.dont\_vibrate:  
            if (item.isChecked()) item.setChecked(false);  
            else item.setChecked(true);  
            return true;  
        default:  
            return super.onOptionsItemSelected(item);  
    }  
}

Jika Anda tidak mengatur status diberi tanda cek dengan cara ini, maka status item (kotak centang atau tombol radio) yang terlihat tidak akan berubah bila pengguna memilihnya. Bila Anda telah mengatur status, aktivitas akan menjaga status diberi tanda cek suatu item sehingga bila nanti pengguna membuka menu, status diberi tanda cek yang Anda atur akan terlihat.