|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio Minggu-2 | 80 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Bisa memahami android studio sesuai dengan yang ada di video.

2. Bisa menggunakan dan mengganti atribut sesuai dengan yang ada di video.

3. Bisa mengganti id pada komponen yang telah dibuat, untuk mempermudah pemanggilan komponen pada saat dibutuhkan kelak (pada video terdapat kotaj number, button, dan text view yang diganti nama id nya). Menggantinya terdapat di attributes pada bagian atas sendiri (kotakan id).

4. Hint (pada attributes), untuk membuat tulisan seperti dibelakang layar dari kotakannya (jadi pada saat kotakannya diisi, maka tulisan tersebut akan hilang dan tergantikan oleh tulisan yang diisikan di hint tadi. Bila tidak ada tulisannya, maka yang muncul adalah tulisan yang sudah dibuat di hint tadi pada kotakannya (pada video kotakan number, diisi dengan tulisan “isi dengan angka (bil 1 dan 2)” (bil 1 untuk kotakan number yang pertama dan bil 2 untuk kotakan number yang ada di bawahnya (yang kedua)) di hint nya)).

5. Bisa mengganti isi text saat ditampilkan di layar di dalam textview (pada bagain text di attributes).

6. TextSize, untuk mengubah ukuran dari tulisan yang sudah dibuat sebelumnya (tampilan pada layar). Bisa mengubah baik lewat attributes (design) maupun code.

7. Bisa menampilkan hasil code yang telah dibuat di android studio dan ditampilakn pada emulator yang sudah di download sebelumnya (nox player, emulatornya sudah harus tersambung terlebih dahulu).

8. Run App (berlogo warna hijau segitiga menghadap ke kanan di bagian atas (di sebelah tampilan sambungan device)), untuk menjalankan/memulai code yang sudah dibuat sebelumnya (memutar dan menunjukkan hasilnya di device yang sudah ditujukan (misal di nox, maka akan langsung muncul di layar nox tersebut (bila success/berhasil) namun bila gagal/error, maka tidak akan muncul di tampilan layar nox dan menampilkan bagian errornya yang mana dan apa)).

9. Stop (berlogo warna merah persegi/kotak di bagian atas (dekat dengan run app)), untuk memberhentikan aplikasi/project yang sedang dijalankan.

10. Declared Attributes, berisikan pengubahan dari komponen attributes secara mendasar (seperti text, id, layout width, layout height, dan lain-lain).

11. Layout, untuk mengatur pengait (cantolan) handle dari komponen yang sebelumnya sudah dimasukkan (seperti komponen kotakan number, text view, button, dan lain-lain).

12. Transfroms, untuk mengubah rotasi atau arah dari suatu komponen yang telah dimasukkan (misal seperti text yang dapat diubah arahnya, misal ke arah yang sebaliknya, dan masih banyak lagi lainnya).

13. Common Attributes, berisikan pengubahan tambahan (misal seperti sudah menambahkan inputType, hint, dan lain-lain akan tampil di sini).

14. All Attributes, berisikan seluruh pengaturan atau pegubahan dari attribute (seperti mengubah warna background, warna text, mengubah font dari teks (fonFamily), dan masih banyak lagi lainnya).

15. Bisa membuat dan mengerti event sesuai dengan yang ada di video.

16. Event, suatu kejadian bila suatu komponen di klik, maka akan dapat menampilkan sesuatu lainnya sesuai dengan yang sudah dibuat sebelumnya.

17. OnClick, akan menampilkan suatu kejadian dengan di klik terlebih dahulu (komponennya di klik terlebih dahulu, seperti button hitung pada video).

18. Bisa membuat suatu kejadian (event) dengan menekan tombol button hitung (dengan dua cara).

* Pertama dengan menambahkan OnClick di dalam activity\_main.xml di bagian code (pada button) (bila ada error maka akan diberi tahu (tanda seru merah di sebelah kanan pojok atas) yang kemudian di sebelah kiri baris code yang error tersebut maka akan memunculkan seperti tombol bergambar tanda seru merah, dan bila di klik akan menampilkan beragam macam pilihan untuk membantu error tersebut (pada video di klik yang ada tombol seperti bohlam lampu, Create ‘hitung(View)’ in ‘MainActivity’)). Setelah itu, maka akan terbuat void atau method (public void hitung (View view) {}) di main activity. (Yang ini nantinya yang akan dipakai).
* Kedua dengan cara menutup (diberi command) pada void/method yang pertama telah dibuat dan menghilangkan OnClick yang terdapat di code button. Lalu mendeklarasikan komponen yang akan diberikan event tersebut, dengan cara mengetikkan nama komponennya, untuk class (Button) dan memberikannya variabel (pada video bernamakan hitung) di main activity. Lalu di dalam void onCreate (di bawah setelah code nya selesai (setContentView)) menuliskan variabel yang telah dibuat sebelumnya. Kemudian menambahkan findViewById(R.id.btnHitung); (R untuk res (resource), id untuk menunjukkan bahwa akan memanggil id, dan btnHitung merupakan id dari suatu komponen yang telah diubah sebelumnya (id dari button hitung)). Lalu menambahkan hitung.setOnClickListener(new View.OnClickListener(), yang berfungsi untuk membuat void baru di dalam suatu void (void onClick).

19. Toast, untuk menampilkan suatu pesan saat berada di layar tampilan (menuliskan kata atau kalimat di toastnya diletakkan di bagian text (seperti pada video yang menampilkan “Menampilkan pesan”, setelah button hitungnya di klik)).

20. Untuk membuat tampilan, maka harus mendeklarasikannya terlebih dahulu (seperti EditText dan TextView), yang setelahnya akan dimasukkan ke dalam void hitung.

21. FindViewById, untuk menempatkan suatu komponen yang sudah memiliki id masing-masing (dipanggilnya menggunakan ini).

22. Bisa mengambil suatu nilai dari EditText dan menampilkannya di TextView sesuai dengan yang ada di video. (Pada saat ditampilkan di emulator (layar), maka hasilnya, bila mengisikan angka pada kotakan komponen number yang pertama (yang atas) dan mengklik tombol button hitung, maka pada text view (untuk teks atau tulisan) di bawahnya akan ikut berubah sesuai dengan angka yang dimasukkan di kotakan komponen number tersebut).

23. Set, untuk meletakkan atau mengisi suatu hasil (pada video digunakan seperti setText).

24. Get, untuk mendapatkan hasil dari suatu hasil yang telah diambil atau diisi oleh set sebelumnya (pada video seperti getText).

25. Menggunakan double atau integer (int) untuk melakukan operator penjumlahan (karena string tidak dapat digunakan untuk menghitung).

26. Bisa mengubah dari suatu variabel yang awalnya string, lalu kemudian menjadi int (integer) atau double.

27. Int a = Integer.parseInt(bil\_1.getText().toString()); , untuk mengubah suatu variabel dari yang awalnya string, lalu dapat berubah menjadi integer (int). (Untuk yang b juga sama seperti yang a, namun bil\_1 nya diganti menjadi bil\_2). Hasil angka yang dikeluarkan oleh integer (int) ialah bulat (1).

28. Membuat dua variabel integer ataupun double untuk menggunakannya di operasi perhitungan nantinya (int a dan int b). Dan membuat int c untuk menambahkan kedua variabel tersebut (a + b). Tidak hanya dapat ditambahkan (penjumlahan (+)), namun juga dapat digunakan untuk penguranan (-), perkalian (\*), dan pembagian (/). Dan mengeluarkan hasilnya dengan hasil.setText(c);.

29. Bisa membuat dan membetulkan error pada saat pembuatan kalkulator, serta menampilkannya di emulator.

30. Double a = Double.parseDouble(bil\_1.getText().toString()); , untuk mengubah suatu variabel dari yang awalnya string, lalu dapat berubah menjadi double. (Untuk yang b juga sama seperti yang a, namun bil\_1 nya diganti menjadi bil\_2). Hasil angka yang dikeluarkan oleh double ialah desimal (1.0) dan untuk mengaturnya, dapat di klik di dalam attributes (search (bergambar seperti kaca pembesar) number, kemudian mengubah centang kotakan numberDecimal yang awalnya tidak tercentang (false) menjadi tercentang (true), (di klik, di centang manual di kotakannya)). Bisa juga bila mangubahnya lewat code, maka yang diubah pada bagian inputType (diganti menjadi numberDecimal). Mengubah numberDecimal nya untuk kedua kotakan number (agar sama-sama desimal).

31. Bila menggunakan double untuk variabelnya, maka seluruh variabel lainnya juga harus double (mulai dari a, b, dan c (untuk hasil)). Sebaliknya pula, bilamana menggunakan integer (int), maka seluruh variabel lainnya juga harus integer (int). Karena bila ada yang menggunakan double bercampur dengan integer (int), maka akan terjadi error dan tidak akan dapat dijalankan (run).

32. Constraint, batasan untuk mencegah error dari sebuah isi komponen, yang bilamana belum terisi, maka akan memunculkan peringatan bahwa kotakan tersebut harus diisi terlebih dahulu.

33. Menggunakan if, else if, dan else (sesuai dengan yang ada di video) untuk membuat constraint (batasan).

34. Bila tidak diberikan constraint (batasan), maka akan terjadi error seperti aplikasi yang tiba-tiba berhenti dengan sendirinya (seperti kalkulator telah berhenti, buka aplikasi lagi).

35. Class, sebuah wadah atau tempat untuk membedakan tiap code yang akan dituliskan. Berisikan method atau function yang berbeda isinya satu sama lain, namun tetap berada di satu class (wadah) yang sama. Terdapat tiga (3) buah class (Public, Private, dan Protected), namun untuk kali ini menggunakan Public terlebih dahulu.

36. Return, untuk mengembalikan atau menjalankan suatu variabel yang telah ditentukan sebelumnya terlebih dahulu. Namun untuk hasil akhirnya diletakkan di return ini (misal seperti return a + b, yang digunakan untuk dapat menambahkan atau menjumlahkan hasil dari variabel (double a, double b) yang sebelumnya telah dideklarasikan terlebih dahulu). Bisa juga untuk selain penjumlahan, seperti pengurangan (-), perkalian (\*), dan pembagian (/).

37. Untuk membedakan class dan method (atau function), dengan cara melihat huruf depan dari namanya tersebut. Bila diawali dengan huruf besar ataupun kapital, maka itu termasuk class. Namun bila huruf depan dari namanya tersebut diawali dengan huruf biasa atau huruf kecil (bukan kapital), maka itu termasuk method atau function. Untuk dapat membedakan hal ini, diharuskan (lebih baiknya) membuatnya dengan kedua perbedaan tersebut (agar dapat mengenali dan membedakannya juga dengan lebih mudah).

38. Bisa membuat MainActivity baru sesuai dengan yang ada di video (MainActivity java yang baru dengan nama Kalkulator).

39. Untuk menggunakan class baru yang terdapat di file yang lain (beda file) dengan mendeklarasikannya terlebih dahulu (sesuai dengan yang ada di video). (Kalkulator calc = new Kalkulator();).

40. Untuk memanggil dan menampilkan hasil yang baru saja dibuat (dengan class Kalkluator), dengan double c = calc. ….(a,b);. (Titik-titik tersebut berisikan fungsi dari yang akan dijalankan, misal ingin menjalankan penjumlahan, maka titik-titik tersebut diisi dengan penjumlahan (double c = calc.penjumlahan(a,b);)).

**Saya Belum Mengerti**

1.

2.

3.