|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Android Studio Minggu-3 | 80 |

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

1. Membuat file project counter sesuai dengan yang ada di video (beserta isinya).

2. Menghapus sebuah komponen dengan cara menekan komponen yang ingin dihapus, kemudian klik delete.

3. Untuk menampilkan penampilan layout di design .xml agar menjadi tampilan depan seperti pada saat di run pada emulator memunculkan action bar nya, pilih simbol seperti mata pada pojok kiri atas (bawahnya pengaturan design dan blueprint), kemudian pilih Show System UI.

4. MainActivity.java, file tempat untuk mengetikkan code utamanya.

5. Activity\_main.xml, file tempat untuk membuat tampilan desain dari project android studio yang akan dibuat.

6. Code (di atas pojok kanan), untuk menampilkan codenya (menambahkan elemen yang diketik).

7. Design (di atas pojok kanan), untuk menampilkan layar untuk mengatur tampilannya (namun bukan dengan code, melainkan di klik dan di tarik).

8. Blueprint, untuk mengetahui tata letak dari komponennya agar bisa pas atau baik. Untuk memunculkan blueprint, dapat dimunculkan pada tombol di sebelah kiri yang bergambar seperti lapisan kertas yang menumpuk (yang di sebelah kanannya ada gambar seperti rotate hp). Dengan cara di klik lalu bisa dipilih, mau yang layer halaman putih saja, blueprint nya saja, maupun tampil keduanya secara bersamaan.

9. + (plus), di bagian kanan bawah (dibawahnya tombol bergambar tangan), untuk memperbesar layar tampilan dari design.

10. - (minus), di bagian kanan bawah (dibawahnya tombol + (plus)), untuk memperkecil layar tampilan dari design.

11. Run App (berlogo segitiga warna hijau 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)).

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

13. Constraint Layout, untuk menghubungkan (handle) suatu bulatan yang ada di tiap sisi komponen (atas, bawah, kanan, dan kiri). Cantolan atau pegangan (handle) tersebut agar suatu komponen tidak berpindah tempat ke tempat yang bukan semestinya diarahkan (karena bila tidak dihubungkan maka pada saat di run, suatu komponen tersebut akan otomatis langsung berpindah kearah pojok kiri atas).

14. Bisa mengatur atribut komponen sesuai dengan yang ada di video.

15. Button, suatu komponen berbentuk kotakan (lebih ke arah persegi panjang). (Pada video berguna untuk menampilkan hasil counter angka (angkanya bertambag keatas dan bisa diturunkan), penjumlahan pada operasi hitung, dan penampilan pilihan isi pada spinner).

16. Horizontally, untuk mengatur arah handle suatu komponen menjadi langsung ke tengah dan hanya ke arah kanan dan kiri.

17. Vertically, untuk mengatur arah handle suatu komponen menjadi langsung ke tengah dan otomatis langsung terkaitkan dengan bagian atas dan bawahnya. Bila diatasnya terdapat atribut, maka terkaitkan dengan bagian bawah dari attribute yang ada di atasnya.

18. Bisa menghapus tiap handle yang sudah terkait pada tiap sisi dengan cara klik handle yang ingin dihapus, kemudian pilih delete. Bisa juga dengan cara mengarahkan (memilih) handle yang akan dihapus dan bila garis handlenya sudah berubah warna yang awalnya biru menjadi merah, maka handle tersebut sudah siap untuk dihapus (tinggal di klik saja bila sudah berubah warna). (Tapi pada Android Studio saya belum bisa menghapus dengan cara mengarahkannya saja ke handle tersebut (yang kemudian garisnya berubah warna)).

19. Clear All Constraint, untuk menghapus atau menghilangkan semua handle yang telah terpasang atau diatur sebelumnya (tempatnya berada di dekat simbol mata (untuk mengatur tampilan layar pada design sebelum di tampilkan di emulatornya)).

20. Wrap Content, ukuran dari suatu komponen yang mengikuti isi yang ada di dalamnya (misalkan bertambah panjang tulisannya, maka kotakan komponennya akan ikut melebar sesuai dengan isinya). Menggunakannya dengan cara tidak perlu merubah garis apapun yang ada di tengah pengaturan handle yang terdapat di dalam Constraint Widget.

21. Fixed, ukuran dari suatu komponen yang tidak mengikuti isi yang ada di dalamnya (misalkan bertambah panjang tulisannya, maka kotakan komponennya tidak akan ikut melebar (akan menetap) berapapun isi yang ada di dalamnya). Menggunakannya dengan cara mengklik sekali garis yang ada di tengah pengaturan handle yang terdapat di dalam Constraint Widget.

22. Match Constraint, ukuran dari suatu komponen yang tidak mengikuti isi yang ada di dalamnya, namun kotakannya mengikuti lebar dari tampilan layarnya, berapapun isi yang di dalamnya tidak akan memengaruhi lebar kotakan. Menggunakannya dengan cara mengklik dua kali garis yang ada di tengah pengaturan handle yang terdapat di dalam Constraint Widget.

23. Bisa mengatur jarak handle di dalam attributes di bagian constraint widget (diubah di setiap kotakkannya, sesuai dengan arahnya, misal ingin memberi jarak yang ada di sebelah kanan, maka kotakan handle yang ada di sebelah kanan yang diubah nominal angkanya (namun harus dikaitkan terlebih dahulu baru bisa diberi jarak)).

24. Bisa menambahkan berbagai macam attribute yang dibutuhkan sesuai dengan yang ada di video.

25. Text Size, untuk mengubah ukuran dari suatu teks sesuai dengan yang diinginkan (belakang dari angkanya harus diberi sp (satuannya) untuk mengubah ukuran teksnya, misal seperti 36sp).

26. Bisa mengubah dan menambahkan tulisan yang ada di dalam text view, button, kotakan number, dan lainnya. Baik dari design (terletak di attributes bagian text) maupun code.

27. Bisa mengganti warna warna button sesuai dengan yang diinginkan (pada bagian attributes, di background). Warnanya dapat dipilih dari warna yang sudah ada di dalam values, colors.xml (dengan @colors/…., titik-titik tersebut berisikan nama warna yang sudah dicantumkan dalam file colors.xml sebelumnya) dan bisa juga dari warna yang memilih sendiri (dari kotakan pallete warna yang sudah disediakan maupun di bagian atasnya yang dapa di klik sesuai dengan selera menginginkan warna yang mana), bisa juga dengan menuliskan kode heksadesimal dari warnanya (misalkan seperti ini, #FFFFF) lalu klik enter.

28. Menambahkan app:backgroundTint=”@null” dibagian code dari suatu komponen button yang ingin diganti warnanya (untuk menghilangkan warna bawaan dari android studionya, dikarenakan punya saya tidak dapat berubah sebelumnya, setelah diberikan ini baru mau berganti warna).

29. Text Style, untuk menjadikan suatu teks menjadi lebih tebal (bold), miring (italic), maupun normal (pada kotakan disebelah dari tulisan pilihannya ini harus di klik, untuk memakai style tersebut).

30. Rotate (berlogo seperti fungsi layar hp putar otomatis), berada di sebelah kanan dari logo lapisan kertas (layer dan blueprint), untuk mengubah tampilan layar dari project yang telah dibuat (mendemonstrasikannya seperti pada saat di run dan ditampilkan dengan beragam macam layar dari perangkatnya (misalkan menjadi landscape, dan yang lainnya)).

31. Bisa membenarkan letak dari teks yang sebelumnya masih kurang pas (dengan mengatur ukuran margin dari tiap handlenya menjadi nol (0) dan menggunakan wrap content agar menyesuaikan dengan isinya).

32. Bisa mengatur id dari suatu komponen, baik lewat code maupun design (di attributes). (Dapat berguna pada saat pemanggilan komponen tersebut). Untuk memberikan id nya, tidak boleh ada spasi (misal seperti etBil\_1, tidak boleh etBil 1). Menggantinya terdapat di attributes pada bagian atas sendiri (kotakan id).

33. Bisa berhasil membuat counter, yang bila mana mengklik button Counter Up, maka nantinya angka (teks yang bertuliskan angka nol (0)) yang berada di tengah akan naik menjadi angka 1, 2, 3, 4, dan seterusnya. Bila mana mengklik button Counter Down, maka akan mengubah angka yang semulanya berada di angka atas setelah di klik Counter Up, maka angkanya akan turun dan sampai mencapai negatif (-1, -2, -3, -4, dan seterusnya).

34. On Click, akan menampilkan suatu kejadian dengan di klik terlebih dahulu (komponennya di klik terlebih dahulu, lalu akan menampilkan apa). Pada awal setelah menambahkannya, maka akan terjadi error, membenarkannya dengan mengklik logo seperti bohlam lampu namun berwarna merah di samping kiri dari baris yang error (onClick), lalu kemudian klik Create ‘btnUp(View)’ in ‘MainActivity’ (pada error onClick di button btnUp, bila error pada button lain atau attribute dari komponen yang lain, maka namanya akan mengikuti dari id komponen tersebut (bukan lagi btnUp)). Setelah itu, maka akan terbuat void atau method (public void btnUp (View view) {}) di main activity.

35. System.Out.Println, untuk menampilkan pesan atau sesuatu yang telah dibuat dan diatur sebelumnya di code, yang kemudian setelah sesuatu di klik atau diberi apapun yang sesuai dengan arahan sebelumnya, akan memunculkan pesan atau sesuatu tersebut, dan munculnya ada di Logcat (seperti button Counter Up yang diberikan System.Out.Println dan memunculkan tulisan sesuai dengan yang diatur sebelumnya (misal btn UP) dengan mengklik button Counter Up pada saat di run, dan muncul dari tulisannya tersebut berada di Logcat (setiap di klik akan mengeluarkan tulisan atau pesannya)).

36. Logcat, untuk memunculkan segala data dari sebuah project yang sedang dijalankan (di run), bilamana ada error juga bisa dilihat dan ditemukan solusinya lewat sini. (Pada video kali ini berfungsi untuk melihat pesan dari yang sudah dituliskan di System.Out.Println, dengan mengklik sesuatu tombol (button) terlebih dahulu). Berada dibagian bawah sendiri berjejeran dengan beragam hal salah satunya seperti TODO, Problems, dan lainnya.

37. Untuk membuat tampilan, maka harus mendeklarasikannya terlebih dahulu (seperti int count dan TextView), yang setelahnya akan dimasukkan ke dalam void btnUp.

38. Count++, untuk menjadikan sesuatu menjadi naik ke atas (misal dari angka 1 naik ke angka 2 dan seterusnya). Penulisan ini sama seperti penggunaan yang ada di java (karena bahasa android studionya menggunakan java).

39. Count--, untuk menjadikan sesuatu menjadi turun ke bawah (misal dari angka 2 turun ke angka 1 dan seterusnya). Penulisan ini sama seperti penggunaan yang ada di java (karena bahasa android studionya menggunakan java).

40. Int, untuk menambahkan suatu variabel dari yang awalnya string, lalu dapat berubah menjadi integer (int), khususnya digunakan untuk angka. Dan sama seperti double, bilamana sudah menggunakan integer (int), maka jangan dicampurkan dengan double).

41. FindViewById, untuk menempatkan suatu komponen yang sudah memiliki id masing-masing (dipanggilnya menggunakan ini). (Penggunaannya seperti findViewById(R.id.tvHasil); (R untuk res (resource), id untuk menunjukkan bahwa akan memanggil id, dan tvHasil merupakan id dari suatu komponen yang telah diubah sebelumnya (id dari button TextView))).

42. Set Text, untuk mendapatkan teks atau sesuatu yang telah diisikan ke dalamnya dan menampilkannya.

43. Bisa mengubah dari string ke integer (int) menggunakan count+” ”. Bisa juga kata dari count tersebut diubah selain count, seperti hasil (hasil+” “) dan lain sebagainya sesuai dengan yang diinginkan.

44. Load, bila setelah membuat code, maka hasil yang akan ditampilkannya di lewatkan function load() dan agar dapat digunakan, harus dimasukkan ke dalam function onCreate bawaan dari MainActivity.java dari android studionya (di bawahnya, buka langsung di dalamnya, namun masih satu tutup kurung dengan onCreate).

45. Linear Layout, untuk menjadikan layer dari tampilan otomatis bertingkat (bersap-sap) dan otomatis menjadi rapi (seperti pada saat mengatur suatu komponen lewat handle yang dikaitkan dengan bagian bawah dari komponen (bila komponen tersebut berada di bawahnya) yang ada diatasnya dan mengatur marginnya sama rata). Mengubahnya dari code activity\_main.xml dibagian codenya pada bagian atas sendiri. Yang bertuliskan android.support.constraint.ConstraintLayout diganti menjadi Linear Layout. Maka penutup bagian bawahnya langsung otomatis ikut berubah juga mengikuti bagian atasnya. Setelah itu otomatis tampilan layernya akan tersusun rapi dan seperti sudah bergaris-garis (sudah tertata).

46. Android:orientation, untuk menjadikan garis yang menjadikan tersusun rapinya layer dengan ini (pilihannya terdadpat vertical dan horizontal, pada video menggunakan vertical).

47. Number (Decimal), komponen yang berupa sebuah kotakan yang mengisikan suatu angka (bisa angka biasa (bulat, misal seperti 1) maupun desimal (seperti 1.5)). Pengisiannya hanya dikhususkan untuk angka.

48. Untuk menjadikan margin semua komponennya sama, maka mengklik salah satu komponennya (dari atas ke bawah saja biar lebih mudah), kemudian sambil menekan tombol shift, dan mengarahkannya ke bawah atau bisa juga di klik satu-satu per komponennya. Lalu bila mengubah marginnya, maka semua komponennya akan diberikan margin yang sama juga.

49. Margin, untuk memberikan jarak pada suatu komponen, baik dari jarak bagian atas, bawah, kanan, maupun kiri. Untuk memberikan margin, maka angka dibelakangnya (satuannya) harus diberikan db (berbeda dengan text size yang diberikan sp).

50. Layout\_gravity, untuk mengubah tata letak dari suatu teks (misalkan dari center (tengah), right (kanan), kiri (left), dan masih banyak lagi lainnya).

51. Declared Attributes, semua attribute yang telah diberikan atau diatur sebelumnya akan masuk ke dalam sini (jadi lebih mudah bila ingin mengganti dari suatu attribute yang sebelumnya sudah ditambahkan).

52. Dapat membuat project kalkulator sesuai dengan yang ada di video dan menyelesaikan seluruh buttonnya (pengurangan, perkalian, dan pembagian). Bisa membuatnya sesuai dengan button per fungsinya (misal seperti button penjumlahan, maka bila dua angka dari kotakan number (decimal) dimasukkan, maka akan menambahkan jumlah dari kedua angka tersebut. Berlaku hal yang sama juga untuk button pengurangan (untuk mengurangi), perkalian (untuk mengalikan), dan pembagian (untuk membagi)).

53. Double, untuk mengubah suatu variabel dari yang awalnya string, lalu dapat berubah menjadi double (penggunaannya harus diikuti dengan parseDouble dan bila sudah menggunakan double, maka jangan dicampurkan dengan integer (int)).

54. Bisa memperbaiki error bila saat suatu projectnya tiba-tiba menutup sendiri (force close), seperti Kalkulator has stopped ‘Restart app’.

55. Bisa memberikan batasan untuk kotakan number (decimal) nya (agar tidak terjadi error (force close)), dengan menambahkan if (etBil\_1.getText().toString().equals(“ ”)) || (etBil\_2.getText().toString().equals(“ ”)), yang lalu bawahnya diberi Toast (bila angka dari komponen kotakan number (decimal) nya kosong atau tidak diisi, maka akan memunculkan pesan yang dituliskan di dalam toast tersebut (pada video bertuliskan Ada Bilangan Yang Kosong)). Dan setelahnya diberi else yang diiisikan code dari coding operasi sebelumnya (yang terdapat deklarasi etBilnya beserta penambahan dan hasilnya), bila kotakan number (decimal) nya diisikan, maka akan menjalankan code dari else nya.

56. Or (||), untuk dua code yang dijadikan satu, misal code a dan code b yang dijadikan satu untuk diberikan sesuatu seperti code yang lainnya lagi. (Seperti pada saat pemberian batas untuk kotakan number (decimal) 1 (etBil\_1) dan number (decimal) 2 (etBil\_2) yang akan diberikan toast bila keduanya belum terisi atau masih kosong).

57. 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 “Ada Bilangan Yang Kosong”, setelah button yang sudah diberikan ini di klik)). (Bisa mengisikan toast dengan otomatis (saat setelah di klik tabnya maka akan mengeluarkan isi dari toastnya) dengan mengklik tab pada keyboard).

58. Bisa menggunakan blueprint sesuai dengan yang ada di video.

59. Bisa memasukkan gambar ke dalam layer android studio sesuai dengan yang ada di video.

60. Bisa menggunakan spinner sesuai dengan yang ada di video.

61. Pada saat memasukkan gambar, format gambarnya png, namun bisa juga jpg, dan nama dari filenya harus menggunakan huruf kecil semua dan tidak ada spasinya. Memasukkannya pertama di dalam folder dari project yang sedang dibuat, kemudian pada folder res -> drawable klik kanan dan pilih paste, lalu ada pemilihan tempatnya (dipilih yang bagian atas , yang tidak ada drawable-v24 nya) lalu klik ok, setelah klik ok ada pengecekan untuk nama dari file gambarnya (bila tidak ada huruf besar dan spasinya bila dilanjut klik ok), gambar yang ingin dimasukkan pun lalu masuk ke dalam folder drawable tersebut (pada video gambar thermometer.png yang dimasukkan dan yang akan dibutuhkan nantinya).

62. Menambahkan dan mengganti warna dasar dari layar tampilan android studionya. Menambahkan dan mengganti colorPrimary, colorPrimaryDark, dan colorAccent yang terdapat di colors.xml pada folder values. (Namun pada android studio saya tidak terdapat tiga (3) hal tersebut dan saya mengubahnya dari color yang bernama purple\_200 dan yang lainnya yang merupakan bawaan dari android studionya (sebenarnya sama seperti tiga (3) hal tersebut, namun namanya saja yang berbeda)).

63. Mencari warna untuk mengganti warna pada layar tampilan di web <https://www.materialpallete.com>. Karena disana sudah ada warna yang sesuai untuk setiap turunan warnanya, dari yang utama, terang, maupun gelap.

64. ImageView, untuk menambahkan gambar ke dalam layer android studio (dengan mengklik dan menahan dan ImageView dan menariknya ke dalam layer project yang sedang dibuat). Kemudian memilih gambarnya yang terdapat di folder project (gambar yang sudah dimasukkan sebelumnya ke dalam sini). Setelah memilih file gambar yang diinginkan lalu klik ok. Setelahnya, gambar tersebut juga diberi handle (dikaitkan) dan diatur marginnya sesuai dengan yang diinginkan.

65. Bisa membuat konversi suhu (sesuai dengan yang ada di video) namun baru bisa sampai pemilihan mau diganti dari apa ke apa (belum bisa sampai menghitungnya).

66. Spinner, untuk memilih banyak pilihan seperti pada form biasanya (yang bilamana di klik awal dari kalimatnya maka akan mengeluarkan banyak pilihan lainnya yang dapat dipilih). Ada segitiga kecil mengarah ke bawah pada bagian pojok kanan dari kotakannya.

67. Pada saat awal membuat tampilan untuk konversi suhu, mengaturnya menggunakan blueprint dan bukan layer designnya.

68. Garis yang dapat dinaikkan maupun diturunkan untuk membuat margin dari suatu komponen tidak terlalu dekat (mepet) maupun di renggangkan ada di sebelah kiri dari pengaturan margin dan handel yang ada di sebelah kiri. (Seperti pada video yang mendekatkan (ke titik nol (0)) jarak margin pada teks nol (0)).

69. Memberikan pilihan isi dari spinner bisa lewat code yang berada di values pada folder strings.xml (diketikkan di dalam <resources> setelah string name dari app nya <string name=”app\_name”>Konversi Suhu</string>). Diketikkan <string-array name=”pilihan”>, lalu bawahnya diketikkan <item>….</item> (titik-titik tersebut diisikan nama dari pilihan yang akan dimasukkan, misal seperti Celcius to Reamur pada video, dan masih banyak lagi pilihan celcius diubah ke apa saja).

70. Entries, untuk dapat mengambil nilai dari string-array yang telah dibuat sebelumnya (itemnya). Dimasukkan atau ditampilkan dalam kotakan spinner yang terdapat pada layar (yang akan ditampilkan).

71. String, untuk code yang berisikan teks atau kalimat menggunakan ini dan bukannya integer (int) maupun double.

72. GetSelectedItem, untuk mendapatkan item yang sudah dipilih atau diberikan ini sebelumnya.

73. Bisa juga menambahkan dari dalam code nya yang ada di dalam MainActivity.java. (Membuat function tersendiri untuk isi spinnernya, yang didalamnya diisikan String[] isi = {Celcius to Reamur, dan seterusnya}).

74. Adapter, untuk memasukkan pilihan isi yang sudah dibuat sebelumnya dengan menggunakan function ke dalam isi pilihan spinner. Karena merupakan string, maka perlu membuat array untuk adapternya terlebih dahulu (ArrayAdapter<String> adapter = new ArrayAdapter (context: this, android.R.Layout.simple\_spinner\_item, isi)). Baru kemudian dimasukkan ke dalam adapternya (spinner.setAdapter(adapter)). Hasilnya (tampilan pilihan pada saat di rumenambahka) sama seperti pada saat membuat lewat string.xml.

**Saya Belum Mengerti**

1. Belum bisa mengatur untuk mengganti nama color pada saat penggantian warna dari tampilan android studionya (terdapat colorPrimary, colorPrimaryDark, dan colorAccent yang di android studio saya tidak ada pada awalnya (adanya bernama purple\_200 dan yang lainnya yang sudah dicantumkan atau bawaan dari android studio nya) dan saya ingin mengganti nama tersebut namun belum mengerti caranya (agar tidak bernama purple\_200 dan bisa diganti namanya sesuai dengan yang diinginkan)). Sudah saya coba ganti nama purple\_200 tersebut namun tidak terganti warnanya. Dapat berganti bila warnanya saha yang diubah dan nama dari color nya tetap purple\_200.

2.

3.