

# MVC

---

## 1. Clicker v1

Pada minggu ini kita akan membuat sebuah game sederhana bernama Clicker. Tujuan dari permainan ini adalah menekan tombol “Click Me” secepat mungkin. Skor pemain akan bertambah satu poin tiap kali tombol tersebut ditekan. Skor ini akan dikurangi sebesar 1 poin tiap 200 milidetik. Permainan ini akan menampilkan nilai tertinggi yang pernah dicapai pada sesi permainan tersebut. Contoh jalannya permainan ini dapat dilihat pada video “clicker demo.avi”.

Untuk memudahkan anda, anda telah diberikan sebuah kelas bernama Clicker. Kelas ini berisi potongan kode yang berfungsi untuk mengurangi nilai atribut counter tiap *decrementDelay* milidetik. Anda akan perlu memodifikasi kelas ini agar program anda berjalan seperti yang diinginkan

Tugas anda adalah merancang kelas-kelas pada program ini dengan mengikuti prinsip MVC dan mengimplementasikannya. Berikut ini adalah beberapa petunjuk pengerjaan yang perlu anda ikuti :

- Pisahkanlah kelas-kelas yang menjadi model, view, dan controller dalam 3 buah package terpisah (package view, controller, dan model)
- Karena state dari model pada program ini dapat berubah tanpa adanya interaksi dari user (contoh : skor pemain akan berkurang tiap 200 milidetik tanpa perlu adanya interaksi dari user) maka model akan perlu memberitahu view bahwa datanya telah berubah. **Gunakanlah pattern Observer untuk melakukan hal ini!**

Kumpulkanlah file zip dengan nama R1101xyyy.zip yang berisi file jar dan class diagram (dalam format PNG, dibuat dengan ASTAH) dari program anda.

## 2. Clicker v2

Untuk menunjukkan keuntungan dari prinsip MVC, pada bagian ini kita akan mengubah spesifikasi permainan kita. Pada Clicker v2, pada saat pemain menekan tombol “Click Me” maka :

- Bila selisih waktu antara penekanan tombol kali ini dan penekanan sebelumnya adalah antara 750 – 1250 milidetik maka skor pemain akan bertambah 1.
- Bila selisih waktunya di luar batas tersebut maka skor pemain akan menjadi 0.

Contoh jalannya permainan Clicker v2 dapat anda lihat pada video “clicker2 demo.avi”.

Bila anda telah merancang nomor 1 dengan baik maka perubahan yang perlu dilakukan hanyalah :

- Membuat sebuah model baru
- Mengubah model yang digunakan oleh controller

**Anda seharusnya tidak perlu mengubah bagian view dari program ini!**

Berikut ini adalah beberapa petunjuk pengerjaan yang akan sangat membantu pengerjaan nomor ini:

- Anda dapat menggunakan `System.currentTimeMillis` yang pernah ditunjukkan pada kuliah ASD untuk mendapatkan waktu pada saat ini dalam satuan milidetik.
- Anda dapat menggunakan ide yang mirip dengan thread pada bagian 1 untuk mengubah skor pemain menjadi 0 bila pemain belum menekan tombol dalam 1250 milidetik.
- Bila anda telah menyelesaikan soal ini dan mendapati bahwa program anda terkadang berhenti selama beberapa ratus milidetik (terutama pada waktu anda menekan tombol “Click Me”) lalu kembali berjalan dengan normal, maka kemungkinan besar penyebabnya adalah karena program anda melakukan instansiasi objek secara berlebihan. Karena sifat program java yang di-manage memorinya oleh JVM (dipelajari di AOK), maka Java harus menghapus secara periodik objek-objek yang tidak lagi digunakan oleh program anda. Proses ini disebut sebagai garbage collection (GC). Pada saat GC dilakukan, program anda akan dihentikan oleh JVM sampai proses GC selesai.

Di bawah ini adalah contoh-contoh kasus yang menyebabkan proses GC harus sering dilakukan oleh JVM :

- Menginstansiasi sebuah objek secara berkali-kali. Cara yang semestinya dilakukan adalah dengan menginstansiasinya satu kali saja lalu menggunakannya berkali-kali.
- Melakukan konkatenasi objek String secara berulang-ulang. Anda dapat mendapatkan informasi mengenai hal ini dengan mencari “java immutable string” dengan Google.

Kumpulkanlah file zip dengan nama R1102xyyy.zip yang berisi file jar dan class diagram (dalam format PNG, dibuat dengan ASTAH) dari program anda.