

## Soal No 2.

Intro :

Anda diminta membuat aplikasi Type Racer yang mempertandingkan 3 bot untuk mengetik kalimat tertentu.

Alur :

Disediakan template kode berikut jadi silahkan dilengkapi TODOnya agar outputnya sesuai yang diminta.

### Class Typer

```
1  class Typer extends Thread {
2      private String botName, wordsTyped;
3      private double wpm;
4      private TypeRacer typeRacer;
5
6      public Typer(String botName, double wpm, TypeRacer typeRacer) {
7          this.botName = botName;
8          this.wpm = wpm;
9          this.wordsTyped = "";
10         this.typeRacer = typeRacer;
11     }
12
13     public void setBotName(String botName) {
14         this.botName = botName;
15     }
16
17     public void setWpm(int wpm) {
18         this.wpm = wpm;
19     }
20
21     public void addWordTyped(String newWordsTyped) {
22         this.wordsTyped += newWordsTyped + " ";
23     }
24
25     public String getWordsTyped() {
26         return wordsTyped;
27     }
28
29     public String getBotName() {
30         return botName;
31     }
32
33     public double getWpm() {
34         return wpm;
35     }
36
37     @Override
38     public void run() {
39
40         String[] wordsToType = typeRacer.getWordsToType().split(" ");
41
42         // TODO (1)
43
44         // TODO (2)
45
46         this.addWordTyped("(selesai)");
47         // TODO (3)
48     }
49 }
```

TODO 1 : Buatlah variable howLongToType yang memuat waktu yang diperlukan typer untuk menulis 1 kata dalam milisecond

TODO 2 : Buatlah perulangan untuk menambahkan kata dengan method addWordToTyped setelah interval waktu sebanyak howLongToType

TODO 3 : Tambahkan typer yang telah selesai mengetik semua kata ke list typeRaceTabel yang ada di class typeRacer.

## Class Result



```
1  class Result {
2      private String name;
3      private int finishTime;
4
5      public Result(String name, int finishTime) {
6          this.name = name;
7          this.finishTime = finishTime;
8      }
9
10     public String getName() {
11         return name;
12     }
13
14     public void setName(String racerName) {
15         this.name = racerName;
16     }
17
18     public int getFinishTime() {
19         return finishTime;
20     }
21
22     public void setFinishTime(int racerTime) {
23         this.finishTime = racerTime;
24     }
25 }
```

TIDAK ADA TODO

## Class TypeRacer

```
1 class TypeRacer {
2     private String wordsToType;
3     private ArrayList<Typer> raceContestant = new ArrayList<>();
4     private ArrayList<Result> raceStanding = new ArrayList<>();
5
6     public String getWordsToType() {
7         return wordsToType;
8     }
9
10    public ArrayList<Typer> getRaceContestant() {
11        return raceContestant;
12    }
13
14    // Word by Yusuf Syam, Silahkan diubah jika dirasa kurang bijak
15    private String[] wordsToTypeList = { ... };
16
17    public void setNewWordsToType() {
18        Random random = new Random();
19        int angkaRandom = random.nextInt(10);
20        wordsToType = wordsToTypeList[angkaRandom];
21    }
22
23    // TODO (4)
24
25    private void printRaceStanding() {
26        System.out.println("\nKlasemen Akhir Type Racer");
27        System.out.println("=====\n");
28
29        // TODO (5)
30    }
31
32    public void startRace() {
33        // TODO (6)
34
35        // TODO (7)
36
37        // TODO (8)
38    }
39 }
```

TODO 4 : Buat method addResult yang mana digunakan untuk menambahkan typer yang telah selesai (mengetik semua kata), ke dalam list race standing.

TODO 5 : Tampilkan klasemen akhir dari kompetisi, dengan format

```
{posisi}. {nama} = {waktu penyelesaian dalam detik} detik
```

TODO 6 : jalankan kompetisi

TODO 7 : selama semua peserta belum selesai, maka tampilkan

```
Typing Progress ...
=====
- {Nama Kontestan}      => {Text yang telah di ketik}
- {Nama Kontestan}      => {Text yang telah di ketik}
- {Nama Kontestan}      => {Text yang telah di ketik}
```

Setiap 2 Detik

TODO 8 : Tampilkan race standing setelah semua typer selesai.

Untuk data wordTypeToList bisa gunakan contoh berikut :

"Menuju tak terbatas dan melampauinya",

"Kehidupan adalah perjalanan yang penuh dengan lika-liku. Jadikan setiap tantangan sebagai kesempatan untuk tumbuh dan berkembang",

"Cinta sejati adalah ketika dua jiwa saling melengkapi, memberi dukungan dan menginspirasi satu sama lain untuk menjadi yang terbaik",

"Hidup adalah anugerah yang berharga. Nikmati setiap momen dan hargai kebahagiaan sederhana di sekitar kita",

"Perubahan adalah satu-satunya konstanta dalam hidup. Yang bertahan adalah mereka yang dapat beradaptasi dengan fleksibilitas",

"Kebersamaan adalah fondasi yang kuat dalam membangun hubungan yang langgeng dan bermakna",

"Masa depan adalah milik mereka yang memiliki imajinasi, tekad, dan kerja keras untuk mewujudkan visi mereka",

"Ketika kita berbagi dengan orang lain, kita tidak hanya mengurangi beban mereka, tetapi juga memperkaya hati kita sendiri",

"Kesuksesan sejati adalah ketika kita mencapai tujuan kita sambil tetap mempertahankan integritas dan empati terhadap orang lain",

"Rasa syukur adalah kunci untuk mengalami kebahagiaan yang sejati dalam hidup. Mencintai apa yang kita miliki adalah kunci kepuasan yang tak ternilai",

## Class (MAIN)

```
1 public class No2 {
2     public static void main(String[] args) throws InterruptedException {
3         TypeRacer typeRacer = new TypeRacer();
4         typeRacer.setNewWordsToType();
5         System.out.println("|| Text to Type ||");
6         System.out.println("\n" + typeRacer.getWordsToType() + "\n");
7
8         Typer[] typers = new Typer[3];
9
10        typers[0] = new Typer("Bot Mansur", 80, typeRacer);
11        typers[1] = new Typer("Bot ToKu", 72, typeRacer);
12        typers[2] = new Typer("Bot Yukiao", 70, typeRacer);
13
14        typeRacer.getRaceContestant().addAll(Arrays.asList(typers));
15
16        typeRacer.startRace();
17    }
18 }
```

TIDAK ADA TODO