

Worksheet Pertemuan 2 Pekan 4 Algoritma dan Struktur Data

Studi Kasus Larik & String

NIM: 23523195

Nama: Bintang Rizqi Anugrah

A. Membuat Folder Untuk Menyimpan Hasil Praktikum

1. Siapkan folder kosong dengan nama menggunakan NIM masing-masing. Jika folder NIM pada pertemuan sebelumnya mau dimanfaatkan, jangan lupa pindahkan dulu isinya ke folder lain sebagai arsip.
2. Folder ini akan dijadikan tempat untuk menyimpan semua pdf dari worksheet ini beserta file praktikum lainnya.
3. Semua kode program dalam praktikum di pertemuan ini langsung dikerjakan dengan VSCode dan disimpan sebagai projek di folder NIM.

B. Debugging pada VSCode & Percobaan Beberapa Method Penting untuk String pada Java

1. Buatlah projek dengan nama PercobaanString dan disimpan di folder NIM
2. Salinlah kode program pada kotak di bawah ini ke bagian void main projek yang sudah Anda buat: (hati-hati ketika proses salin tempel, karena sangat mungkin ada 'autocorrect' yang justru membuat kesalahan dari aspek kode Java)

```
String kataKata = "";  
// Isilah kataKata dengan NIM dan nama lengkap Anda  
  
char hasil_charAt;  
int hasil_length;  
String hasil_substring_1;  
String hasil_substring_2;  
boolean hasil_contains;  
String hasil_join;  
  
System.out.println("Yang format dicoba sendiri yaa!");
```

3. Lengkapi baris kode program untuk memberi nilai kataKata dengan NIM dan nama lengkap Anda sebagaimana tercantum di baris komentar (baris merah).
4. Sisipkan kode program pada baris sebelum `System.out.println` dengan cara seperti di bawah ini (menggunakan autocomplete dari VSCode sangat membantu ketika melakukan koding, sekaligus bisa mempelajari sintaks baru)

```

38     char hasil_charAt;
39     int hasil_length;
40     String hasil_substring_1;
41     String hasil_substring_2;
42     boolean hasil_contains;
43     String hasil_join;
44
45     hasil_charAt = kataKata.c
46
PROBLEMS (base) zai
charAt(int index) : char      String.charAt(int index) : char
chars() : IntStream
codePointAt(int index) : int
codePointBefore(int index) : int
codePointCount(int beginIndex, int endIndex) : int
codePoints() : IntStream
compareTo(String anotherString) : int
compareToIgnoreCase(String str) : int
concat(String str) : String

```

- Setelah mengetik kode

```
hasil_charAt = kataKata
```

Silakan berhenti, kemudian ketik tanda titik dan karakter c, berhenti lagi, maka akan muncul autocomplete dari VSCode.

- Perhatikan nama-nama method setelah tanda titik, itu semua adalah method untuk tipe data String. Perhatikan juga sisi kanan, itu semua adalah tipe dari setiap method terkait.
- Pilih charAt dan isikan 5 sebagai argumen-nya atau baris kode program akan menjadi:

```

boolean hasil_contains;
String hasil_join;

hasil_charAt = kataKata.charAt(5);

System.out.println("Yang format dicoba sendiri yaa!");
}

```

- Lakukan cara sebagaimana langkah 5 sampai 7 untuk mencoba method-method lainnya dengan parameter seperti tampilan berikut

```
String hasil_join;

hasil_charAt = kataKata.charAt(index: 5);
hasil_length = kataKata.length();
hasil_substring_1 = kataKata.substring(beginIndex: 5);
hasil_substring_2 = kataKata.substring(beginIndex: 5, endIndex: 10);
hasil_contains = kataKata.contains(s: "Tidak Ada");
hasil_join = String.join(kataKata,...elements: " dan ", " tambahan ");

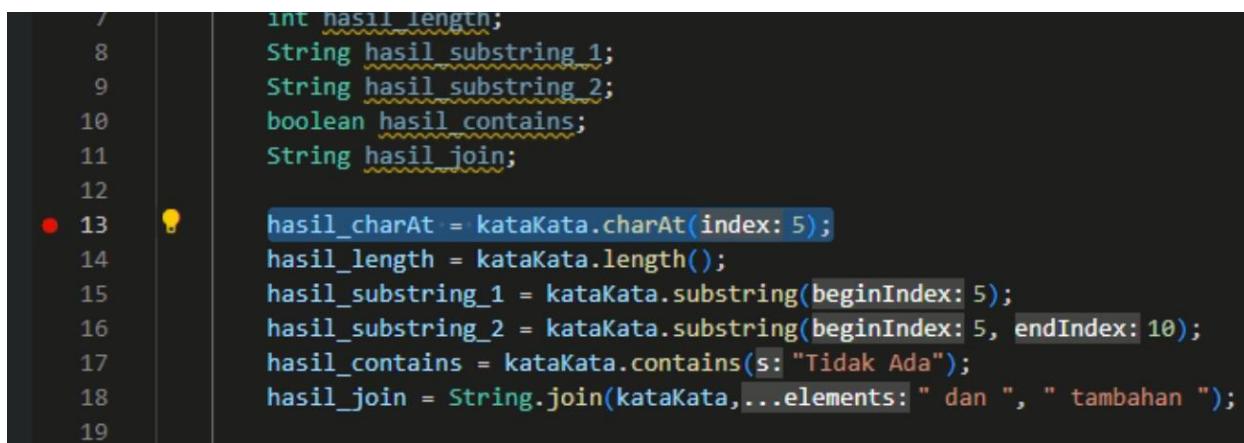
System.out.println(x: "Yang format dicoba sendiri yaa!");
```

Jika muncul kata-kata tertentu seperti “index”, “beginIndex” dan sebagainya dalam parameter method, biarkan saja untuk sementara. Kata-kata itu dimaksudkan untuk membantu pemrogram (*hint*) yang sebenarnya bisa dinonaktifkan.

9. Untuk mengetahui nilai dari setiap variabel yang sudah diisi dengan percobaan beberapa method pada langkah 8, kita tidak perlu mencetaknya ke layar. Kita harus biasakan dengan mendebug-nya karena dalam praktik nyata nanti, akan terlalu banyak variabel yang diproses dalam program yang tidak perlu ditampilkan. Dalam praktik nanti, nilai variabel yang ditampilkan hanyalah hasil akhirnya.

Perhatikan baik-baik langkah 11 berikut setelah dipastikan tidak ada kesalahan dalam program Anda.

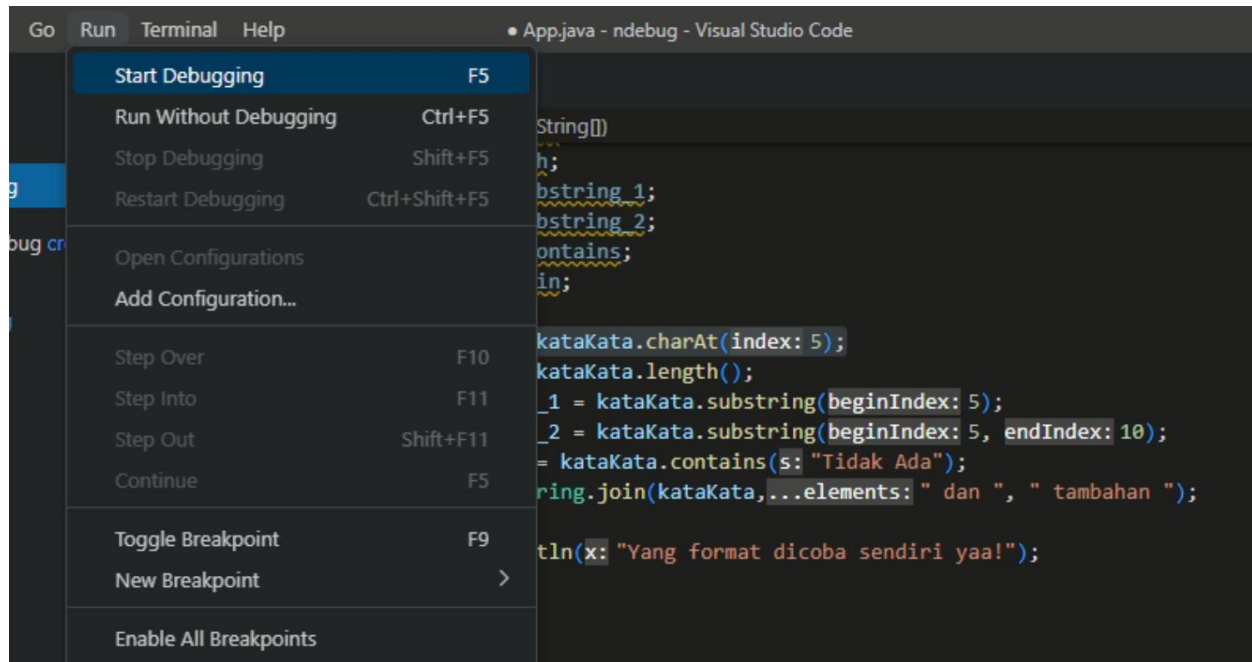
10. Buatlah *breakpoint* untuk proses *debugging* dengan cara arahkan kursor di sebelah kiri nomor baris pernyataan `hasil_charAt = kataKata.charAt(5);` kemudian klik, sehingga muncul tanda noktah seperti gambar di bawah ini:



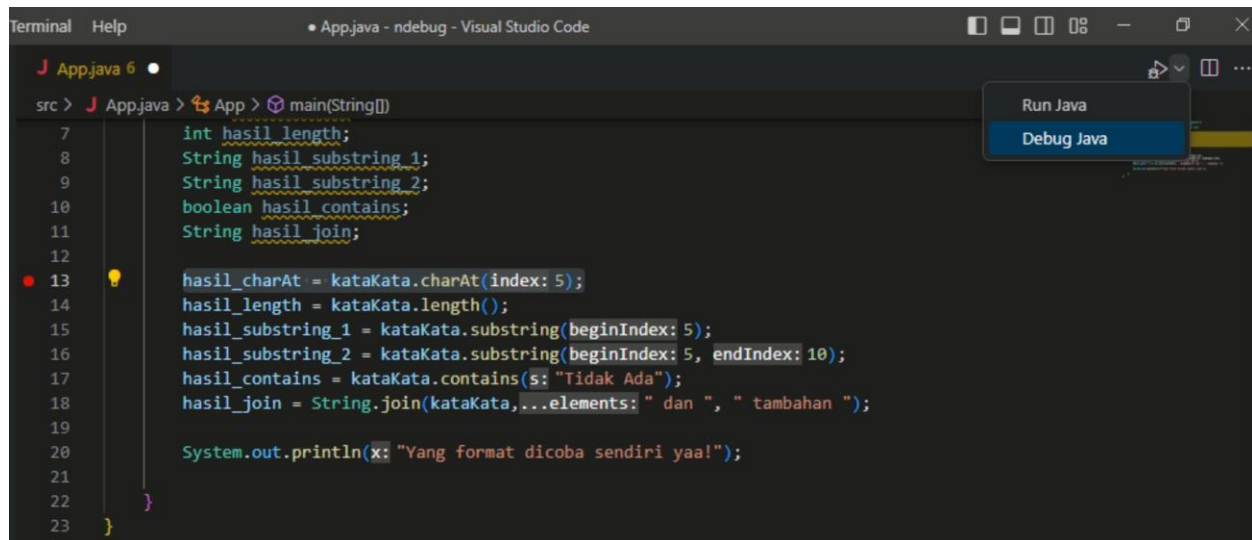
Breakpoint adalah posisi baris yang menjadi pembatas proses *debugging* untuk berhenti. Dalam proses *debugging* yang sebenarnya, menghentikan proses di posisi *breakpoint* ini dimaksudkan untuk memberi kesempatan memeriksa nilai-nilai variabel yang dapat mengarahkan ditemukannya *bug*.

Tentu saja kalau di latihan ini, tujuannya bukan untuk menemukan *bug* tetapi untuk melatih agar kita dapat melakukan proses *debugging* jika suatu saat membutuhkan.

11. Pilih menu **Run | Start debugging** seperti gambar berikut:




Dapat juga dengan klik tanda panah ke bawah di sebelah kanan tombol Run, kemudian pilih **Debug Java** seperti gambar di bawah ini:



- ```
Terminal Help App.java - ndebug - Visual Studio Code

J App.java 6 x
src > J App.java > App > main(String[])
7 int hasil_length;
8 String hasil_substring_1;
9 String hasil_substring_2;
10 boolean hasil_contains;
11 String hasil_join;
12
13 hasil_charAt = kataKata.charAt(index: 5); kataKata = "965240102 Zainudin Zukhri"
14 hasil_length = kataKata.length();
15 hasil_substring_1 = kataKata.substring(beginIndex: 5);
16 hasil_substring_2 = kataKata.substring(beginIndex: 5, endIndex: 10);
17 hasil_contains = kataKata.contains(s: "Tidak Ada");
18 hasil_join = String.join(kataKata,...elements: " dan ", " tambahan ");
19
20 System.out.println(x: "Yang format dicoba sendiri yaa!");
21
22 }
23 }
```

- 
- File Edit Selection View Go Run Terminal Help
- App.java - ndebug - Visual Studio Code
- RUN AND DEBUG ▶ No Configurations
- VARIABLES
- Local
- args: String[0]@8
- > kataKata: "965240102 Zainudin Zuhri"
- WATCH
- src > App.java > App > main(String[])
- ```
7 int hasil_length;  
8 String hasil_substring_1;  
9 String hasil_substring_2;  
10 boolean hasil_contains;  
11 String hasil_join;  
12  
13 hasil_charAt = kataKata.charAt(  
14 hasil_length = kataKata.length(  
15 hasil_substring_1 = kataKata.su  
16 hasil_substring_2 = kataKata.su  
17 hasil_contains = kataKata.conta  
18 hasil_join = String.join(kataKa
```

14. Perhatikan isi bagian VARIABLES, sampai dengan baris yang dieksekusi (di contoh ini baris nomor 13) baru nampak isi nilai variabel kataKata adalah "965240102 Zainudin Zukhri". Isi variabel tersebut adalah hasil eksekusi dari langkah nomor 2 pada baris yang pertama (berwarna merah). Tentu saja kalau variabel tersebut diisi dengan NIM & Nama masing-masing, nilai variabel akan berbeda.
15. Tanda panah akan bergeser ke bawah (di contoh ini ke baris nomor 14) jika Anda klik tombol **Step Over** atau  yang menunjukkan kalau baris nomor 13 sudah dieksekusi.

Amati perubahan yang terjadi pada bagian VARIABLES, seharusnya akan muncul variabel dengan nilainya sebagaimana baris pernyataan pada nomor 13.
16. Ulangi lagi klik **Step Over**, sampai semua variabel muncul di bagian VARIABLES.
17. Aturlah agar semua variabel di bagian VARIABLES tampak, kemudian salin tangkapan layar pada bagian bagian VARIABLES ke dalam kotak di bawah ini

✓ VARIABLES

 ✓ Local

 args: String[0]@8

 > kataKata: "Bintang Rizqi Anug..."

 hasil_charAt: 1

 hasil_length: 30

 > hasil_subString_1: "ng Rizqi ..."

 > hasil_subString_2: "ng Ri"

 hasil_contains: false

 > hasil_join: "Bintang Rizqi An..."

18. Perhatikan nilai setiap variabel kemudian buat kesimpulan mengenai kegunaan dari setiap method terkait. Masukkan kesimpulan Anda untuk melengkapi tabel di bawah ini(kecuali format). Hati-hati dengan *autocorrect* yang bisa mengganti jenis huruf kecil menjadi besar.

No	Nama Method	Keterangan
1	charAt	Memilih karakter yang berada di sub-string yang dimulai dari 0
2	length	Jumlah karakter yang digunakan (termasuk spasi)
3	Substring_1	Menghapus output sejumlah n dari depan

4	Substring_2	Menghapus output sejumlah n dri depan dan menghapus output sejumlah n dari belakang
5	contains	Memeriksa suatu String apakah mempunyai sub-String yang diberikan di dalam contains atau tidak (jika ada berarti true dan jika tidak berarti false)
6	Join	Menambahkan elements di dalam sub-string

C. Aplikasi Pembuat Singkatan dengan Kata-Kata Bertipe String

1. Buatlah proyek dengan nama SingkatanString pada Folder NIM.
2. Kode di bawah ini

```

/**
 * @author zain
 */
import java.util.Scanner;

public class App {
    public static void main(String[] args) throws Exception {
        String huruf;
        boolean lagi;
        boolean tadiSpasi;

        tadiSpasi = true;
        String singkatan;

        Scanner input = new Scanner(System.in);
        singkatan = " ";
        lagi = true;
        while (lagi) {
            huruf = input.nextLine();
            if (tadiSpasi) {
                singkatan = singkatan + huruf;
            }
            if (huruf.equals(".")) {
                lagi = false;
            }
            if (huruf.equals(" ")) {
                tadiSpasi = true;
            } else {
                tadiSpasi = false;
            }
        }
        System.out.println(singkatan);
    }
}

```

jika di-run hasil runningnya adalah seperti gambar di bawah:


```

m
a
r
i

k
i
t
a

c
o
b
a
.
mkc
PS C:\Users\zain\Downloads\testing\ndebug>

```

Program pembuat singkatan dengan masukan kata tetapi dituliskan huruf demi huruf yang diakhiri dengan tanda titik.

3. Modifikasi kode program di atas agar dapat dimanfaatkan untuk membuat singkatan berdasarkan masukan sebagaimana tampilan berikut ini:

```

PROGRAM PEMBUAT INISIAL
=====
Masukan nama lengkap (akhiri dengan titik):
Zainudin Zukhri.
ZZ

```

```

PROGRAM PEMBUAT INISIAL
=====
Masukan nama lengkap (akhiri dengan titik):
Ahmad Munasir Raf'i Pratama.
AMRP

```

4. Simpan proyek Anda.

D. Menampilkan Nilai Elemen Larik yang Memenuhi Syarat Tertentu

Buat proyek dengan nama `UangKembalian` yang akan menampilkan uang pecahan yang harus dikembalikan ke pembeli berdasarkan larik bernama `banyakUang` (nilai elemen-elemennya menunjukkan berapa banyak uang pecahan yang bersesuaian). Sementara variasi nominal uang pecahan disimpan di larik `uangPecahan`.

Sebagai contoh lihat tampilan berikut:

Untuk nilai kedua larik di bawah ini

```
int[] banyakUang = {0, 1, 1, 0, 1, 0, 0, 0, 0, 0};  
  
// uang pecahan bisa ditambah/diedit sesuai kebutuhan  
int[] uangPecahan = {500, 1000, 5000, 10000, 20000, 0, 0, 0, 0, 0};
```

Tampilan hasil *running*-nya adalah:

```
run:  
Uang kembalian terdiri dari:  
1 lembar uang pecahan 20000  
1 lembar uang pecahan 5000  
1 lembar uang pecahan 1000
```

Jangan lupa simpan juga fail worksheet ini (yang sudah diisi) dalam bentuk pdf di folder NIM anda, beserta folder / proyek VSCode.

Kompres folder ini sebagai file ZIP kemudian kumpulkan di classroom atau ruang pengumpulan lain di kelas masing-masing.