**Worksheet Pertemuan 2 Pekan 4 Algoritma dan Struktur Data**

**Studi Kasus Larik & String**

**NIM : 23523170**

**Nama : Danendra Farrel Adriansyah**

1. **Membuat Folder Untuk Menyimpan Hasil Praktikum**
2. 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.
3. Folder ini akan dijadikan tempat untuk menyimpan semua pdf dari worksheet ini beserta file praktikum lainnya.
4. Semua kode program dalam praktikum di pertemuan ini langsung dikerjakan dengan VSCode dan disimpan sebagai projek di folder NIM.
5. **Debugging pada VSCode & Percobaan Beberapa Method Penting untuk String pada Java**
6. Buatlah projek dengan nama PercobaanString dan disimpan di folder NIM
7. 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!"); |

1. Lengkapi baris kode program untuk memberi nilai kataKata dengan NIM dan nama lengkap Anda sebagaimana tercantum di baris komentar (baris merah).
2. 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)

Text

Description automatically generated

1. Setelah mengetik kode

hasil\_charAt = kataKata

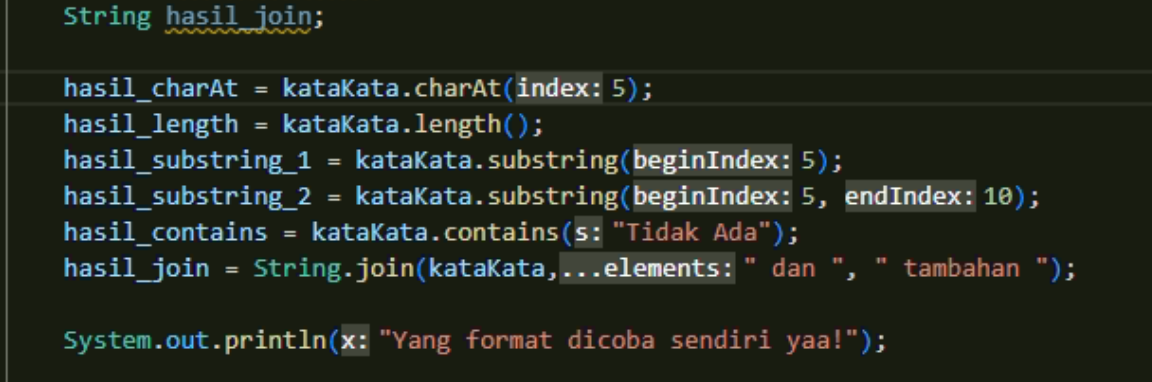
Silakan berhenti, kemudian ketik tanda titik dan karakter c, berhenti lagi, maka akan muncul autocomplete dari VSCode.

1. 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.
2. Pilih charAt dan isikan 5 sebagai argumen-nya atau baris kode program akan menjadi:

Graphical user interface, text, application, email

Description automatically generated

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



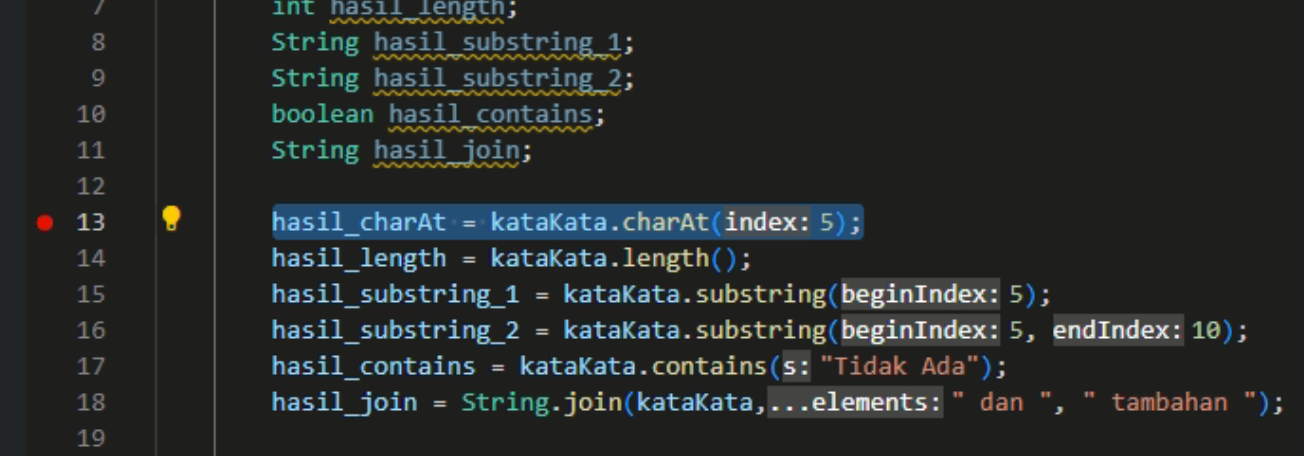
Jika muncul kata-kata tertentu seperti “index”, “beginIndex” dan sebagainya dalam parameter method, biarkan saja untuk sementara. Kata-kata itu dimaksudkan untuk membantu pemrogram (*hint*) yang sebenarnya bisa dinonaktifkan.

1. 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.

1. Buatlah *breakpoint* untuk proses *debugging* dengan cara arahkan kursor di sebelah kiri nomor baris pernyataan hasil\_charAt = kataKata.charAt(5);

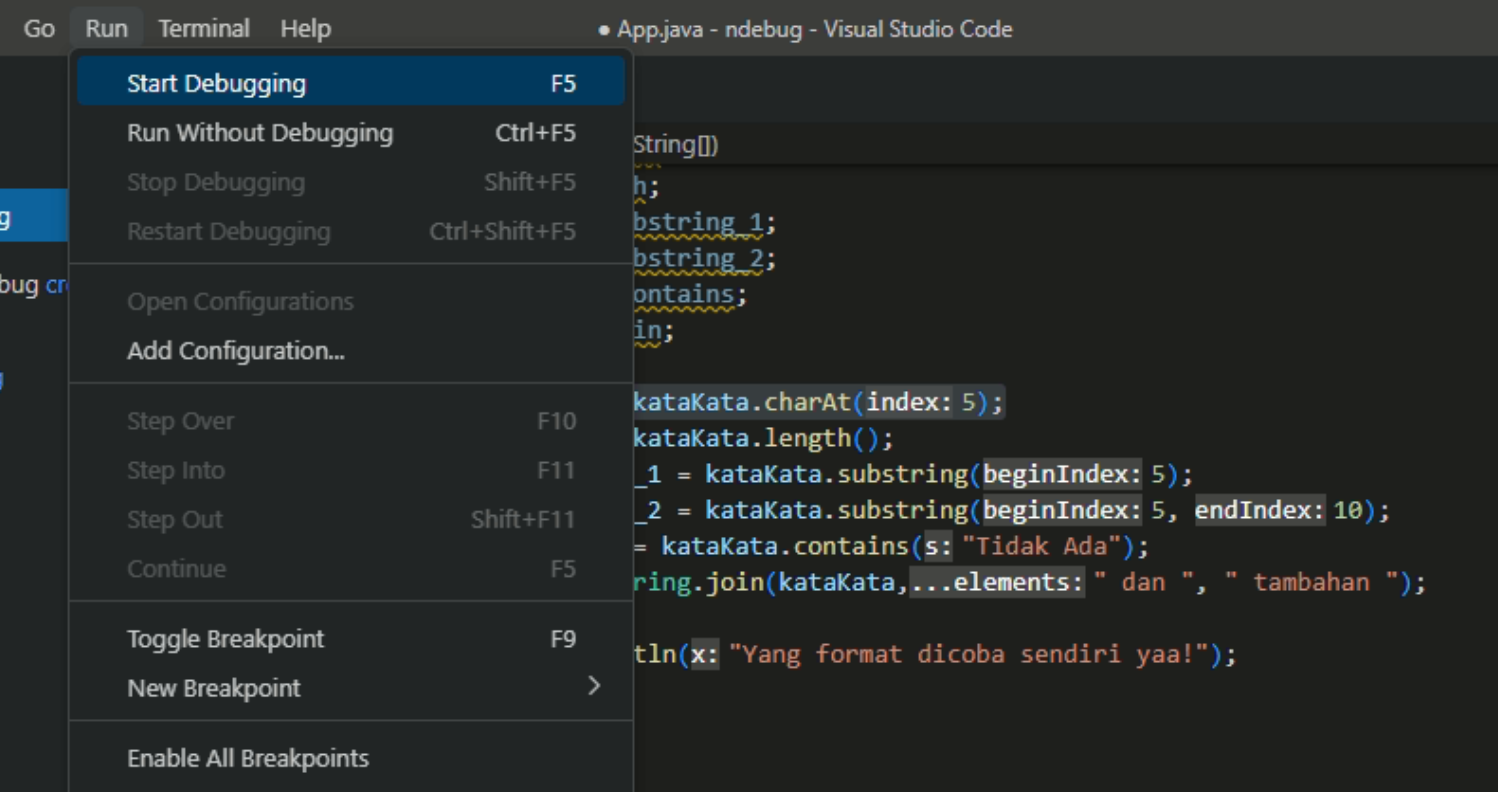
kemudian klik, sehingga muncul tanda noktah seperti gambar di bawah ini:



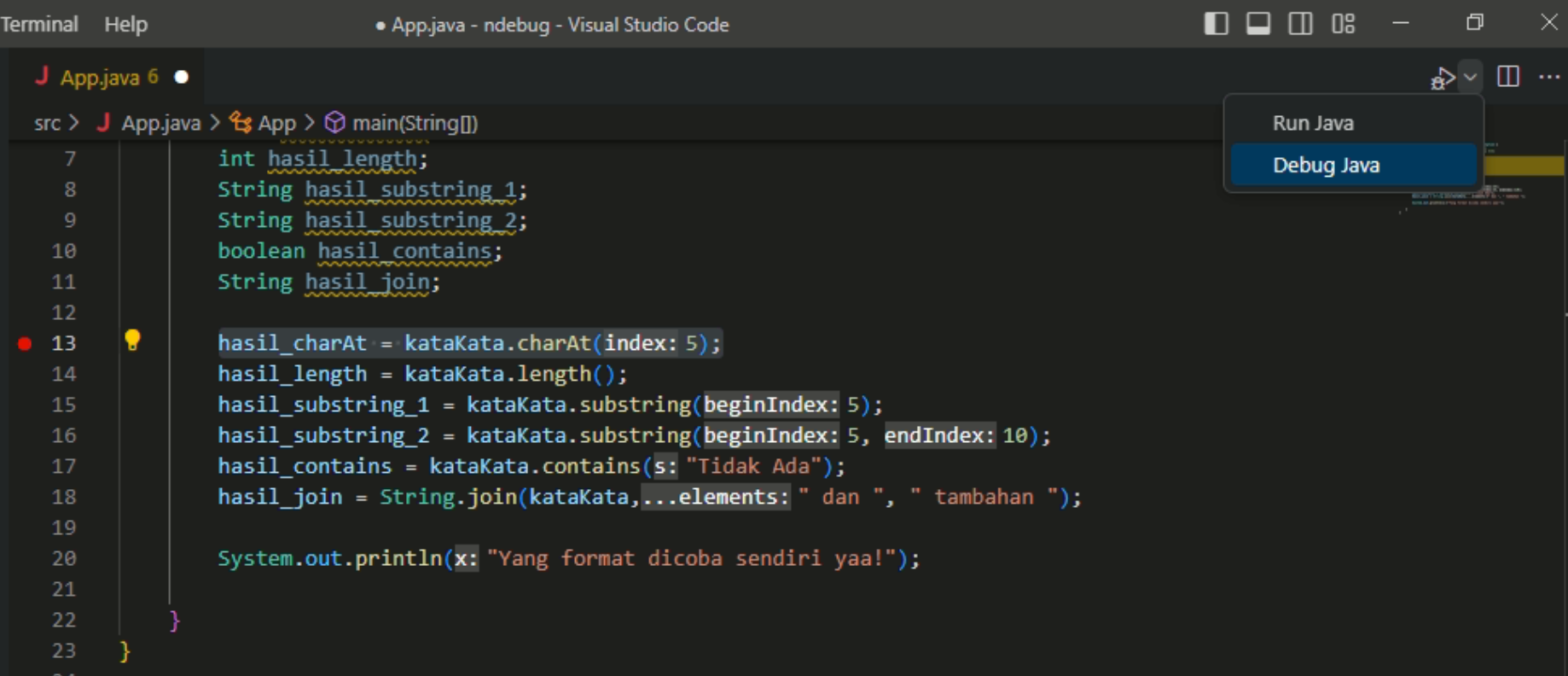
*Breakpoint* adalah posisi baris yang menjadi pembatas proses *debugging* untuk berhenti. Dalam proses *debugging* yang sebenarnya, menghentikan proses di posisi *breakpoint* ini dimaksudkan untuk memberi kesempatan memeriksa nilai-nilai variabel yang dapat mengarahkan ditemukannya *bug*.

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

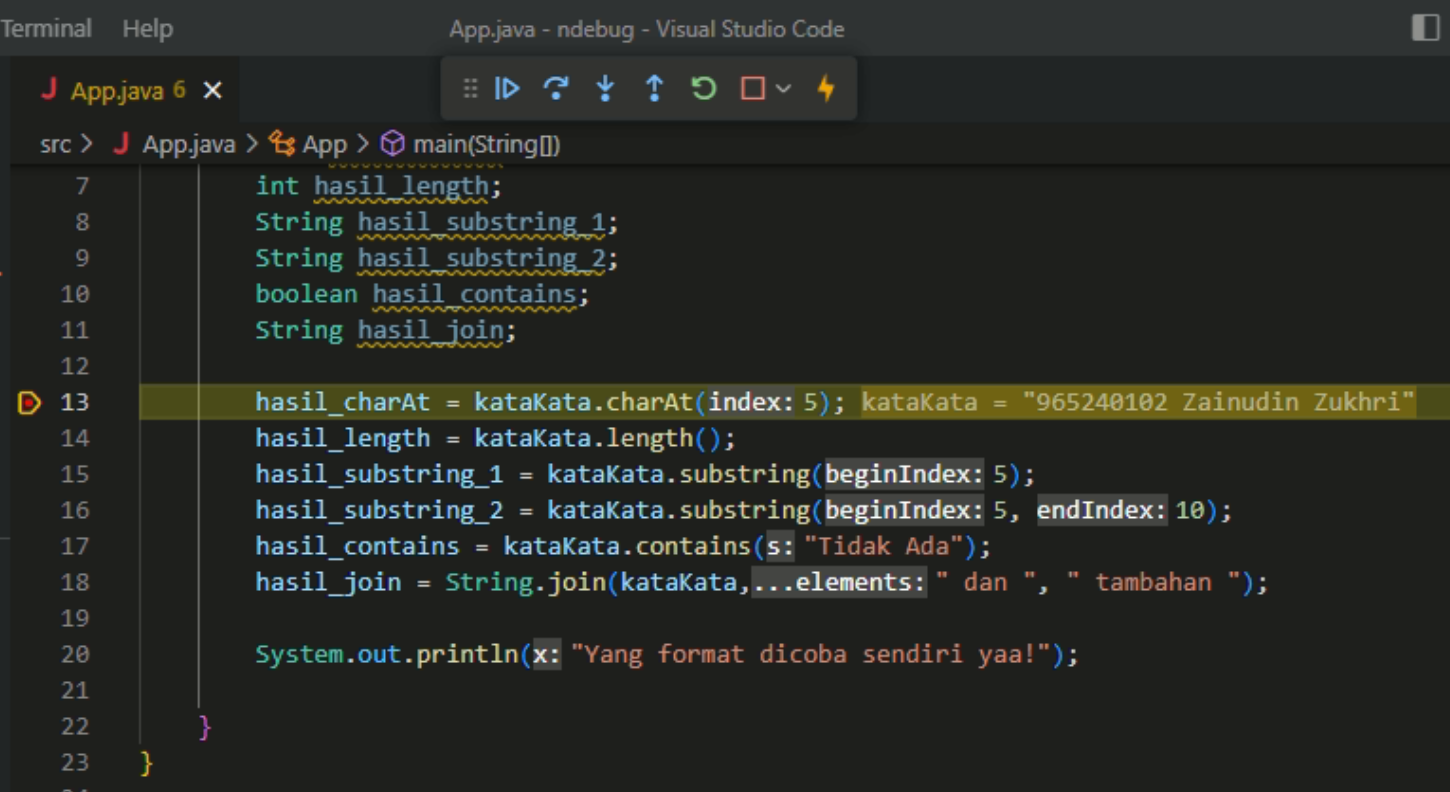
1. Pilih menu **Run | Start debugging** seperti gambar berikut:



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

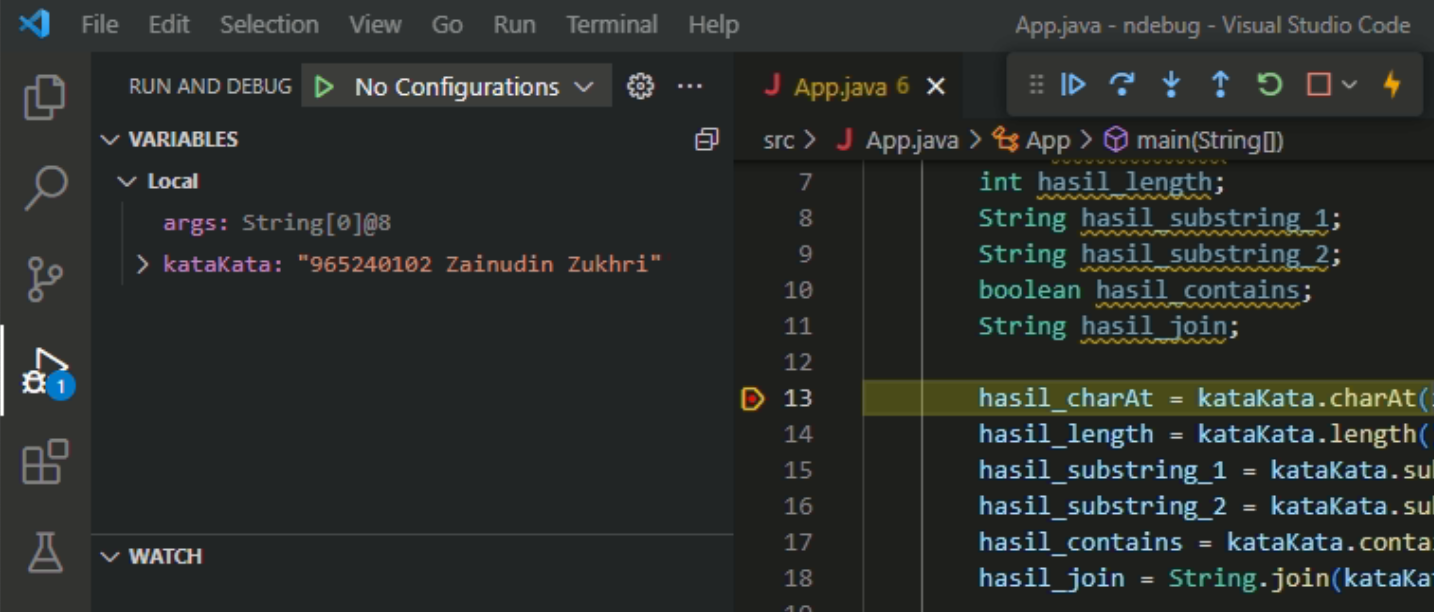


1. Setelah salah satu cara pada nomor 11 dilakukan, pada baris-baris program seharusnya akan muncul tanda panah sekaligus adanya tanda blok atau highligh pada baris di mana posisi breakpoint berada. Tanda itu menunjukkan kalau proses debugging sudah berlangsung. Lihat gambar di bawah ini:



1. Di samping tanda seperti disebutkan pada nomor 12:

* Di atas kode program juga akan muncul galeri tombol-tombol untuk melanjutkan proses debugging
* Di kiri kode program juga akan muncul bagian yang menampilkan nilai-nilai varaibel (VARIABLES) dan bagian untuk mengamati variabel lain yang diinginkan untuk diamati (WATCHES).



1. 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.
2. 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 bagianVARIABLES, seharusnya akan muncul variabel dengan nilainya sebagaimana baris pernyataan pada nomor 13.

1. Ulangi lagi klik **Step Over**, sampai semua variabel muncul di bagianVARIABLES.
2. Aturlah agar semua variabel di bagianVARIABLES tampak, kemudian salin tangkapan layar pada bagian bagianVARIABLES ke dalam kotak di bawah ini

|  |
| --- |
|  |

1. 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 | Mengambil karakter di index tertentu |
| 2 | length | Menghitung jumlah karakter dalam string |
| 3 | Substring 1 | Mengambil substring karakter dari index yang ditentukan yaitu 0 - 5 |
| 4 | Substring 2 | Mengambil substring karakter dari index yang ditentukan yaitu 5 - 10 |
| 5 | contains | Memeriksa apakah string mengandung substring atau tidak |
| 6 | join | Menggabungkan beberapa string menjadi satu |

1. **Aplikasi Pembuat Singkatan dengan Kata-Kata Bertipe String**
2. Buatlah projek dengan nama SingkatanString pada Folder NIM.
3. 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:

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated with medium confidence |

Program pembuat singkatan dengan masukan kata tetapi dituliskan huruf demi huruf yang diakhiri dengan tanda titik (masukkan satu huruf lalu enter diulang-ulang sampai semua huruf pembentuk kata-kata yang akan disingkat sudah di-enter, tutup dengan masukkan titik lalu enter).

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

Text

Description automatically generated

Text

Description automatically generated

1. Simpan projek Anda.
2. **Menampilkan Nilai Elemen Larik yang Memenuhi Syarat Tertentu**

Buat projek 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

Graphical user interface, text, application

Description automatically generated

Tampilan hasil *running*-nya adalah:

Graphical user interface, text, application

Description automatically generated

**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.**