

**LAPORAN PRAKTIKUM TEKNIK PEMROGRAMAN**

**ARRAY, OPERASI STRING DAN ARRAYLIST**



**UIN SUNAN AMPEL  
S U R A B A Y A**

**DOSEN PENGAMPU : DWI ROLLIWATI**

**DISUSUN OLEH :**

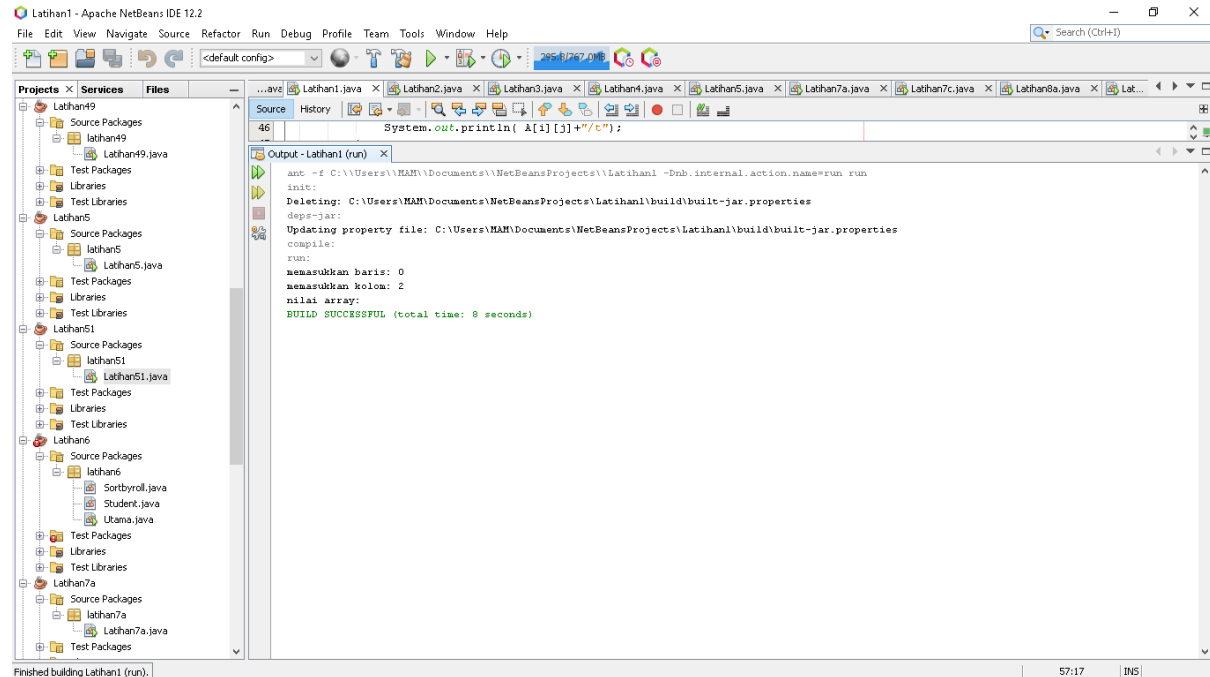
**M.KHOTIBUL UMAM (09020620031)**

**SISTEM INFORMASI**

**FAKULTAS SAINS DAN TEKNOLOGI**

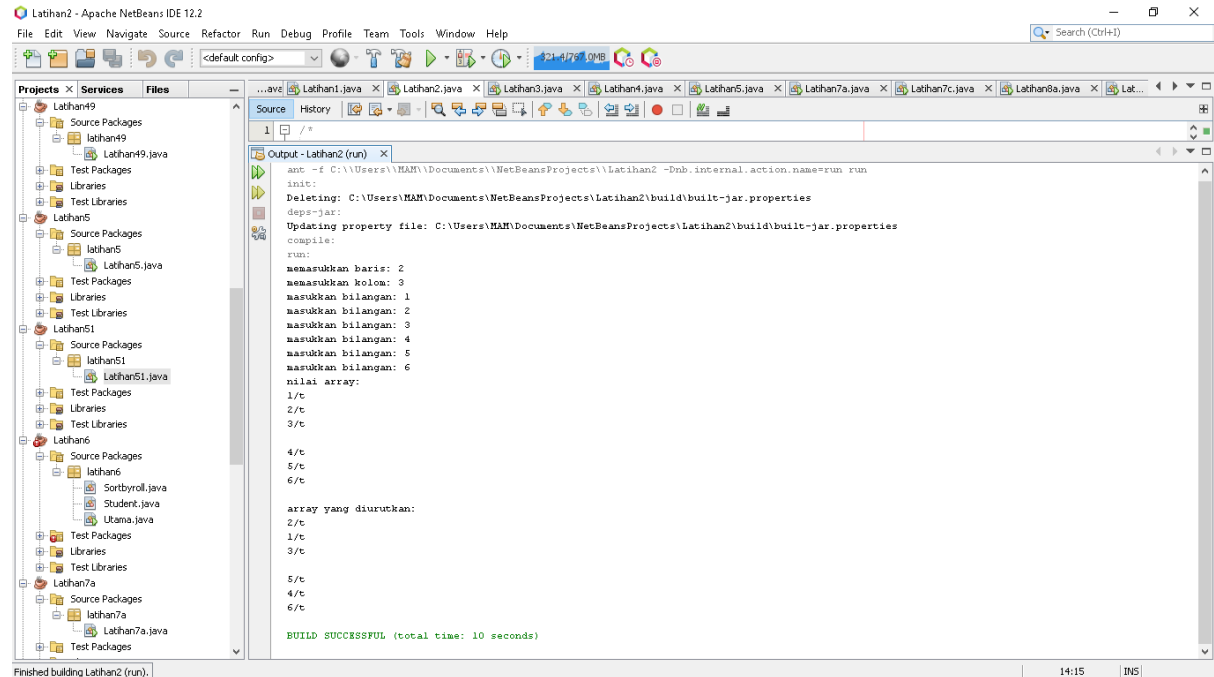
**UNIVERSITAS NEGERI ISLAM SUNAN AMPEL SURABAYA**

# Latihan 1



Hasil dari proses dengan menggunakan Array 2 dimensi dan Array 1 dimensi. Pertama inputkan nilai dari variabel “m” (yang nantinya berfungsi sebagai baris dalam Array 2 dimensi) setelahnya masukkan nilai dari variabel “n” (berfungsi sebagai jumlah kolom dalam Array 2 dimensi). Selanjutnya masukkan isi dari Array 2 dimensi yang jumlah baris dan kolomnya telah terbentuk sesuai dengan nilai pada variabel “m” “n”. selanjutnya print out seluruh nilai Array yang sudah dimasukkan. Selanjutnya proses menyimpan isi Array 2 dimensi ke Array 1 dimensi menggunakan variabel “B” sebagai inisialisasi dan panjang Array 1 dimensi adalah “m\*n”. berikutnya ialah proses pengurutan isi dari Array 1 dimensi dengan menggunakan teknik sorting Standar, dengan proses membandingkan dan mengurutkan setiap isi Array 1 dimensi dengan isi Array 1 dimensi yang lain jika nilai atau yang bisa disebut isi dari Array 1 dimensi tersebut lebih kecil maka otomatis ditaruh di index terdepan, begitu terus hingga bertemu yang paling besar maka akan ditaruh diindex terakhir, jika sudah selesai dengan pengurutan yang telah dilakukan pada Array 1 dimensi setela itu, Array 1 dimensi yang telah diurutkan tersebut disimpan kembali menjadi bentuk Array 2 dimensi. Tahap terakhir yaitu mengeluarkan seluruh isi Array 2 dimensi yang telah diurutkan di proses pengurutan pada Array 1 dimensi.

## Latihan 2



```
Output - Latihan2 (run)
ant -f C:\Users\HAM\Documents\NetBeansProjects\Latihan2 -Dnb.internal.action.name=run run
init:
Deleting: C:\Users\HAM\Documents\NetBeansProjects\Latihan2\build\build-jar.properties
deps-jar:
Updating property file: C:\Users\HAM\Documents\NetBeansProjects\Latihan2\build\build-jar.properties
compile:
run:
masukkan baris: 2
masukkan kolom: 3
masukkan bilangan: 1
masukkan bilangan: 2
masukkan bilangan: 3
masukkan bilangan: 4
masukkan bilangan: 5
masukkan bilangan: 6
nilai array:
1/t
2/t
3/t

4/t
5/t
6/t

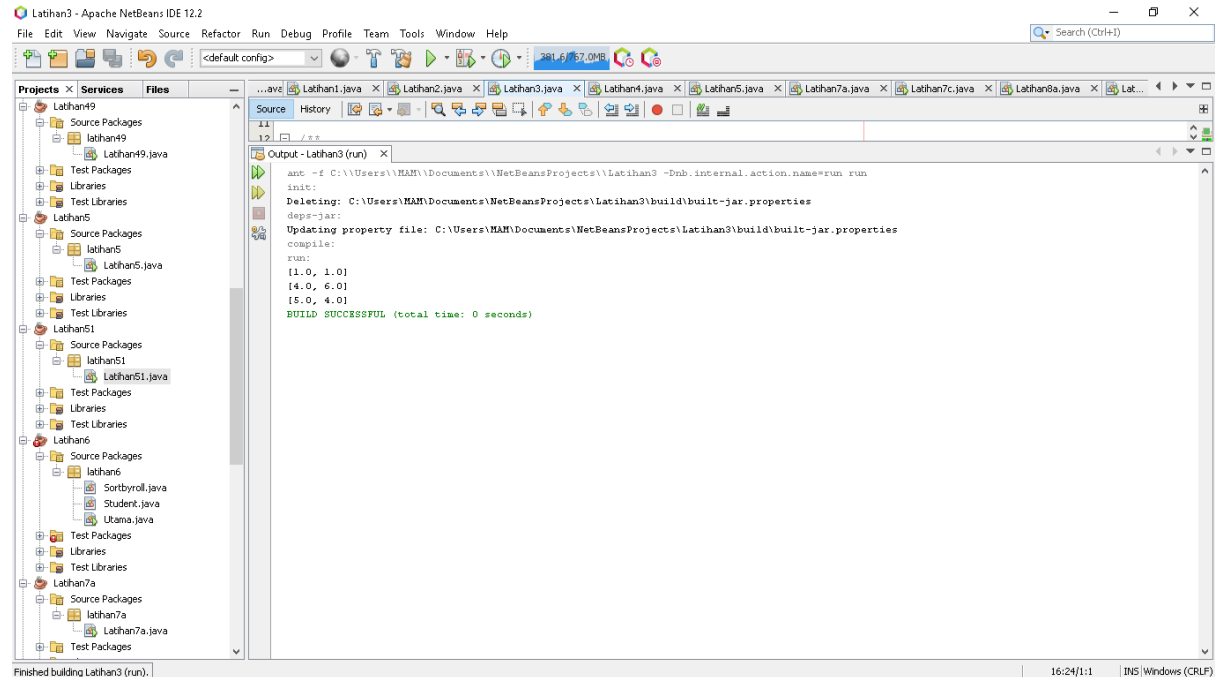
array yang diurutkan:
2/t
1/t
3/t

5/t
4/t
6/t

BUILD SUCCESSFUL (total time: 10 seconds)
```

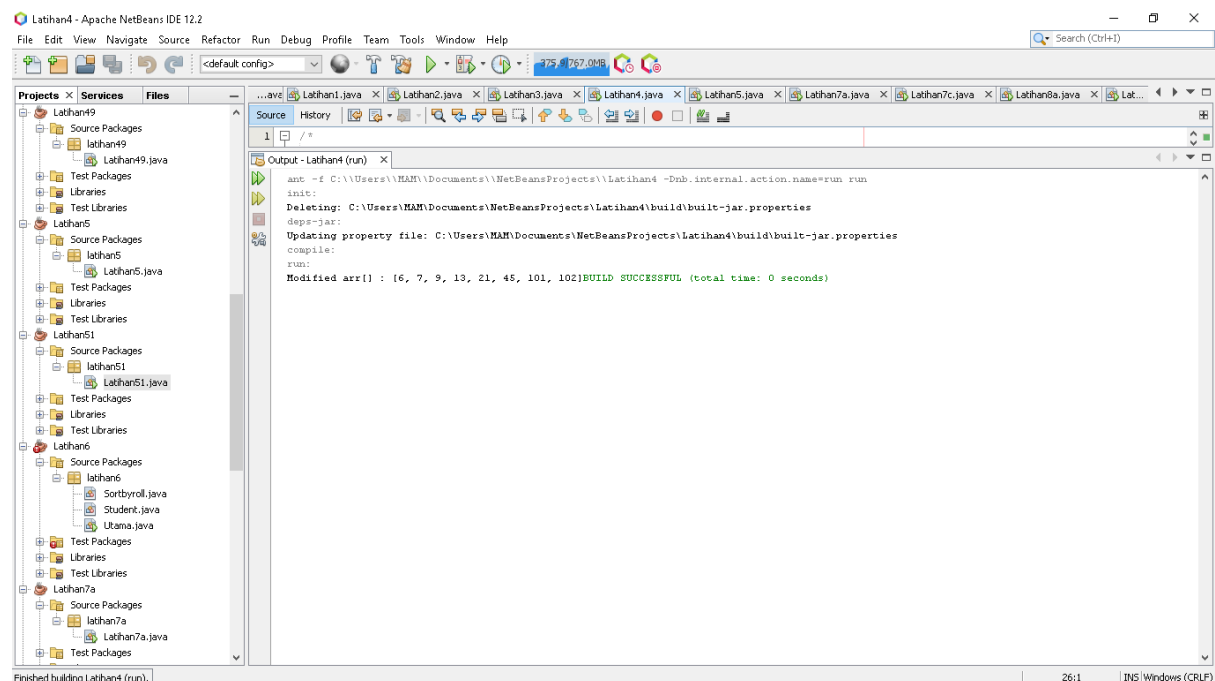
Pengurutan Array 2D secara langsung tanpa merubahnya menjadi Array 1D. dengan cara, pertama mengunci tiap baris dan kolom menggunakan “for” lalu setelah dikunci barulah melakukan looping berbandingan pada setiap baris dan kolom yang dilakukan menggunakan “for”, tahap selanjutnya ialah mengecek looping perbandingan pada setiap baris dan kolom yang sudah dilakukan dengan menggunakan “if” jika benar maka angka terkecil ditempatkan di index terdepan dan angka terbesar ditempatkan di index terakhir. Proses terakhir adalah mengeluarkan Array 2 dimensi yang telah diurutkan pada proses sebelumnya.

## Latihan 3



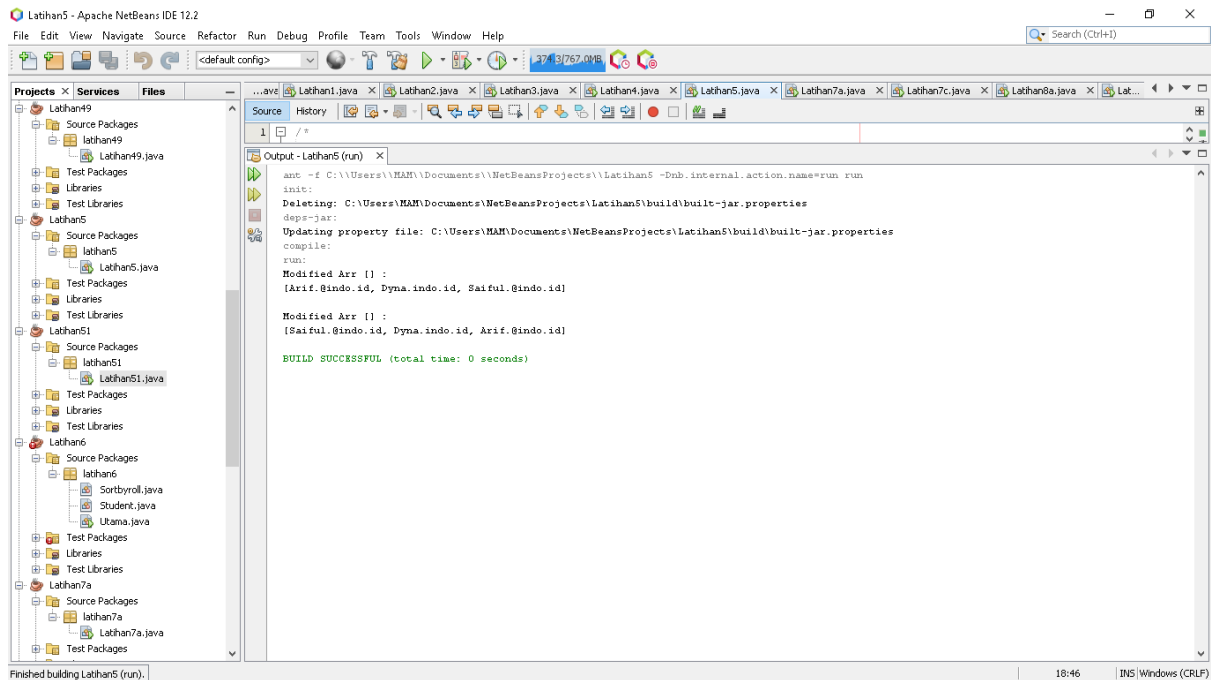
Menggunakan fitur yang ada dilibrary java dan mengurutkan dengan cepat dibanding dengan menggunakan manual tadi. Final yang digunakan pada line 22 dan 26 ini pada saat dijalankan tidak bisa dijalankan lagi karena itu maka dibilang final.

## Latihan 4



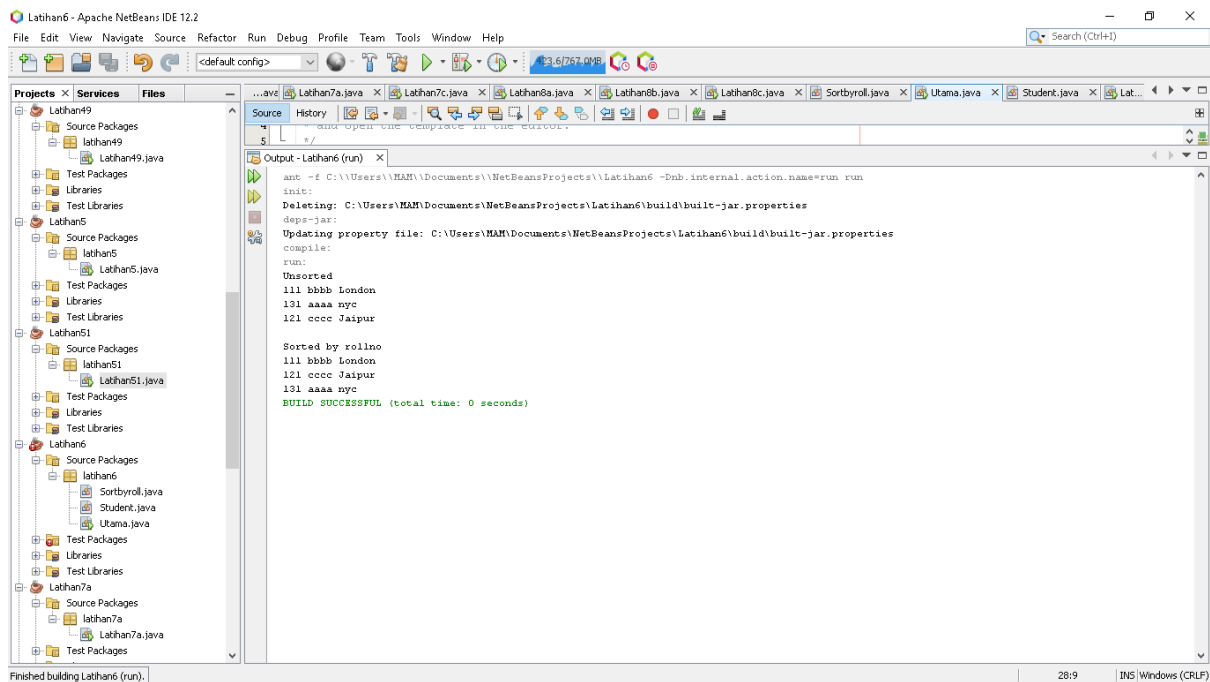
Array sort(arr) pada line 21 adalah me urutkan dengan cara yang simpel menggunakan library java (fitur yang ada dalam java sehingga mempercepat kita dalam melakukan coding) kemudian pada line 22 mengeprint hasil.

## Latihan 5



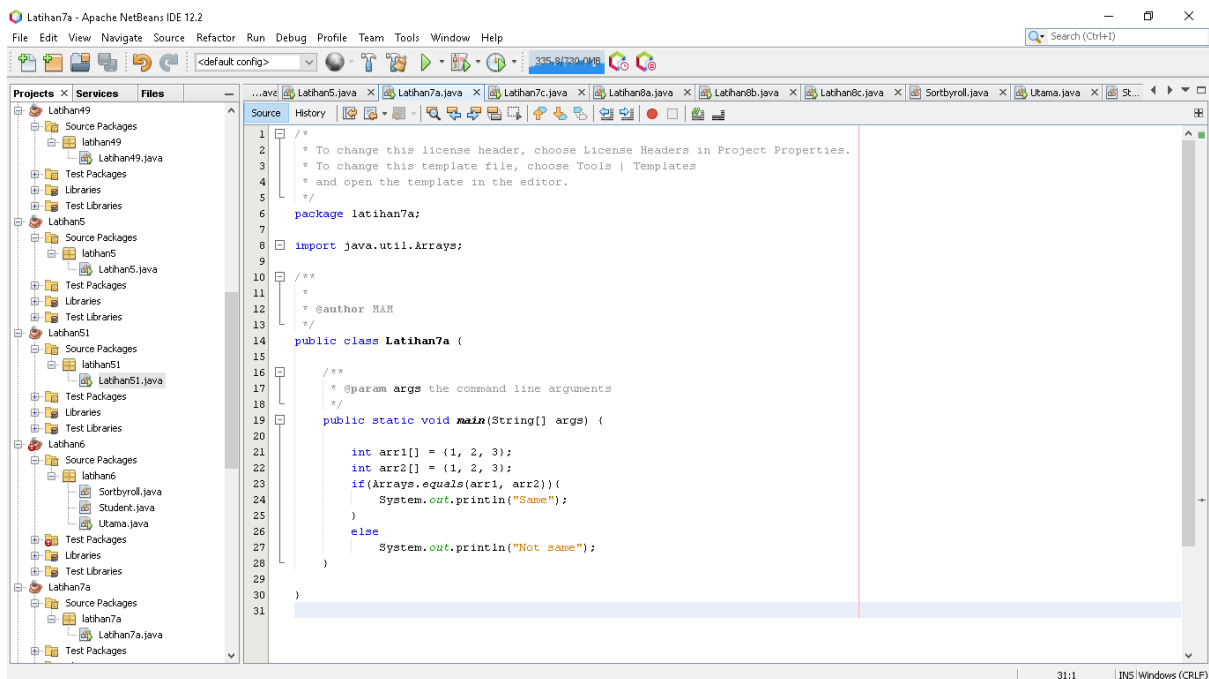
Pertama membuat array String yang sudah kita isi pada codingan [“Khofifah@indo.id, Dwi\\_roll@indo.id, Ashabul kahfi@eng.id”](#) kita akan short secara Ascending menggunakan `Arrays.sort()` kemudian di print hasilnya [Ashabul kahfi@eng.id, Dwi\\_roll@indo.id, Khofifah@indo.id](#). Seteah itu data juga di `Arrays.sort()` namun pada parameter ditambahkan `Collections.reverseOrder()` yang akan membuat pengurutan dari Z ke A (Descending) hasil outputnya [Khofifah@indo.id, Dwi\\_roll@indo.id, Ashabul kahfi@eng.id](#)

## Latihan 6



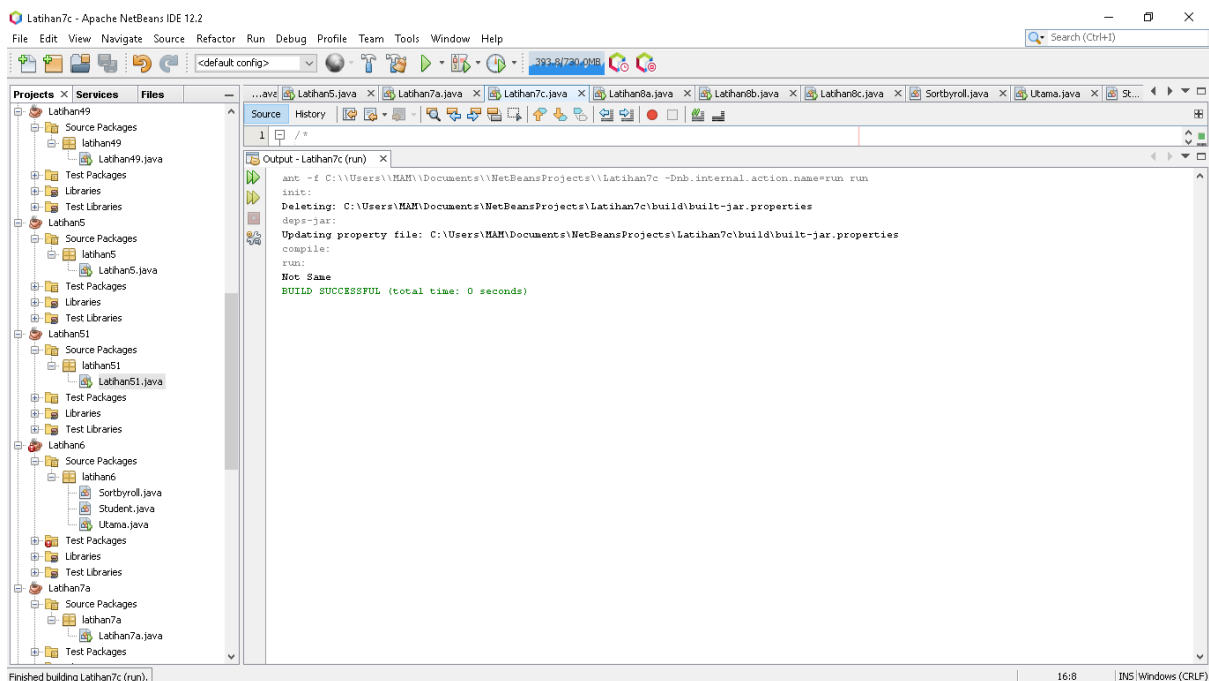
Cara agar bisa membuat beberapa class itu saling berhubungan. Misalkan ada 3 class (yang nantinya kita buat) jika 1 saja subclass (yang bukan utama) itu tidak ada maka class utama tidak bisa jalan atau akan error. Kedua class tersebut dapat saling berinteraksi hingga bisa menjalankan sebuah program pada class utama. Masuk pada codingan, jadi pertama-tama buat 3 class terlebih dahulu, yang pertama membuat class yang isinya untuk menyimpan data rollno, nama, dan alamat. Untuk pemanggilan rollno, nama, dan alamat harus ditambahkan penamaannya agar programnya bisa membedakan mana penamaan data dan pemisalan data yang ingin di panggil (class diberi nama “Student”). Yang kedua membuat class yang isinya untuk membandingkan besar kecilnya angka terlebih dahulu, kemudian diurutkan dari angka terkecil hingga terbesar (class diberi nama “Sortbyroll”). Yang terakhir membuat class sebagai penggerak utama untuk memanggil dari class satu dan dua untuk diproses keseluruhan, kemudian dicetak output unsorted dan sortednya. Pada hasil output terlihat bahwa tanpa sorting, data akan diprinturut sesuai urutan kita beri nilai. Setelah melewati proses sorting maka hasil print akan terurut sesuai dari yang terkecil ke terbesar.

## Latihan 7a



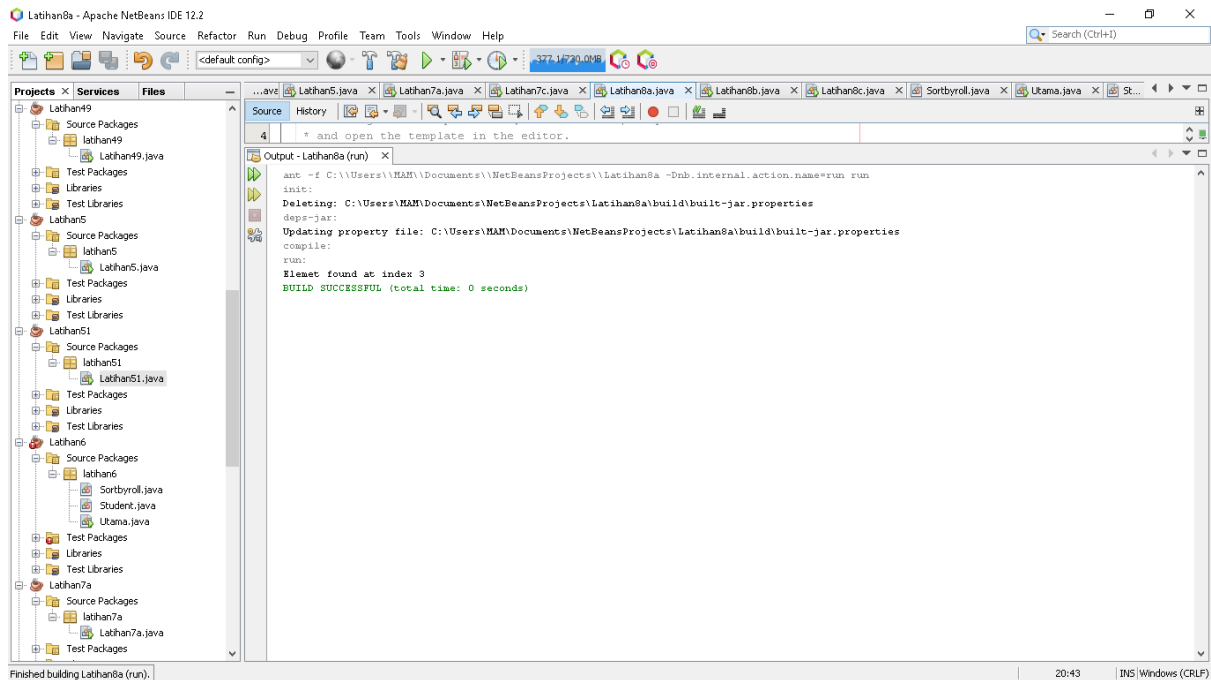
pada program ini prosesnya kita perlu menjalankan loop dan compare elemen dengan satu persatu. `Array.equals()`; disediakan oleh java untuk membandingkan dua Array. Metode `equals()` di class array untuk type primitive yang berbeda ada int, char, etc dan satu untuk type objek yang merupakan basis dari semua kelas di java.

## Latihan 7c



Pada implementasi diatas adalah membandingkan 2 array dengan fitur juga menggunakan (`Arrays.equals(.....)`) disamping adalah contoh implementasi yang akan kamu isi yang sama dengan yang akan kamu bandingkan.

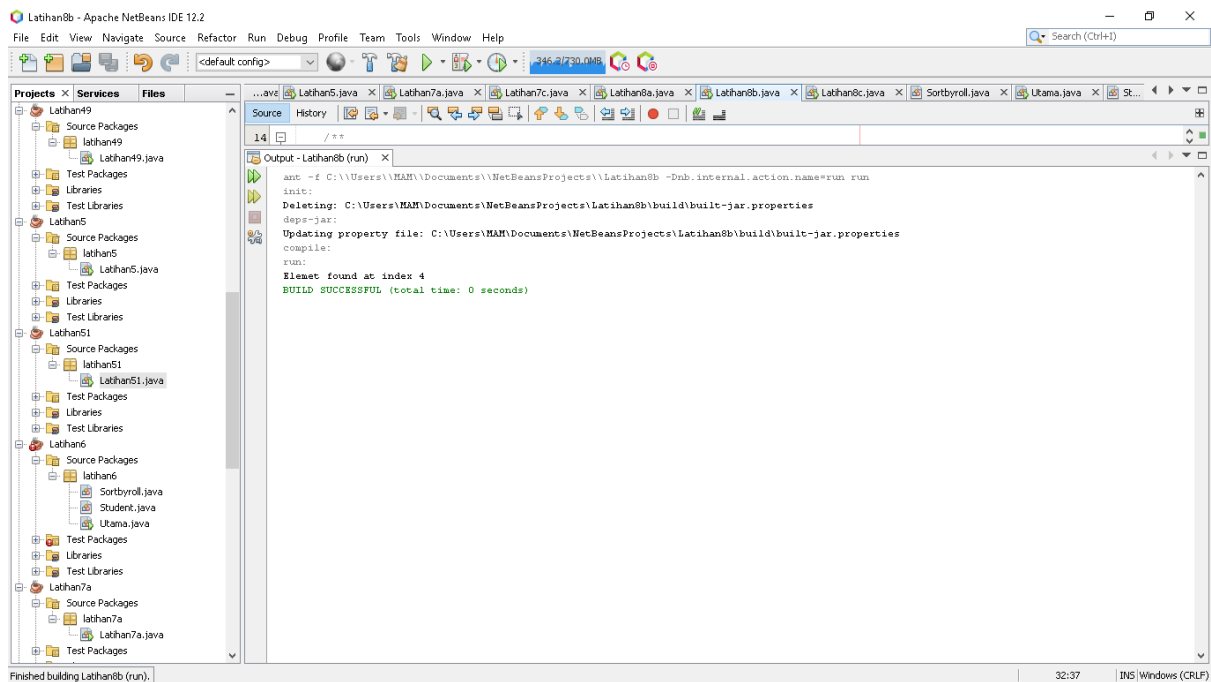
## Latihan 8a



Teknik pencarian data dari array yang paling mudah adalah dengan cara sequential search, dimana data dalam array dibaca satu persatu diurutkan dari index terbesar, maupun sebaliknya. Masuk kecodingan , `int a[] = {3, 34, 5, 91, 100}` jika kita ingin mencari bilangan 91 dalam array tersebut. Maka proses yang terjadi kita mencari dari array index ke-0, yaitu 0, dicocokkan dengan bilangan yang akan dicari, jika tidak sama, maka mencari index berikutnya, pada array index ke-1 dan seterusnya, sampai bilangan yang kita cari ada ditemukan, maka kita keluar dari looping pencarian.

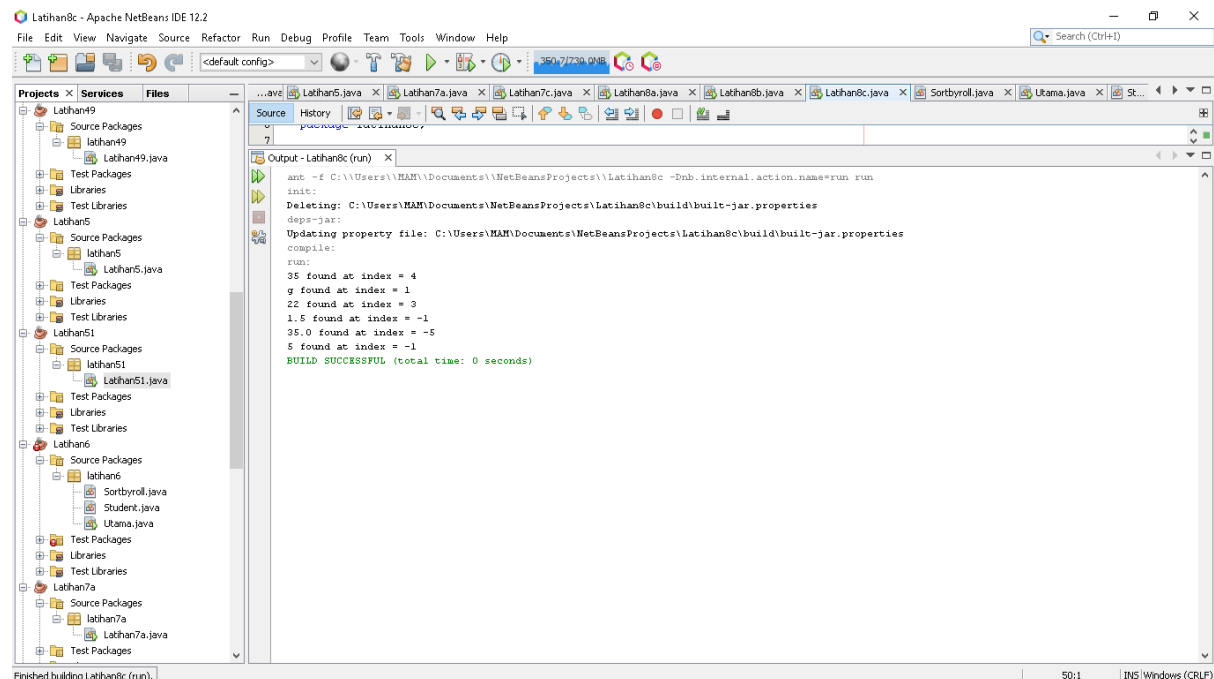


## Latihan 8b



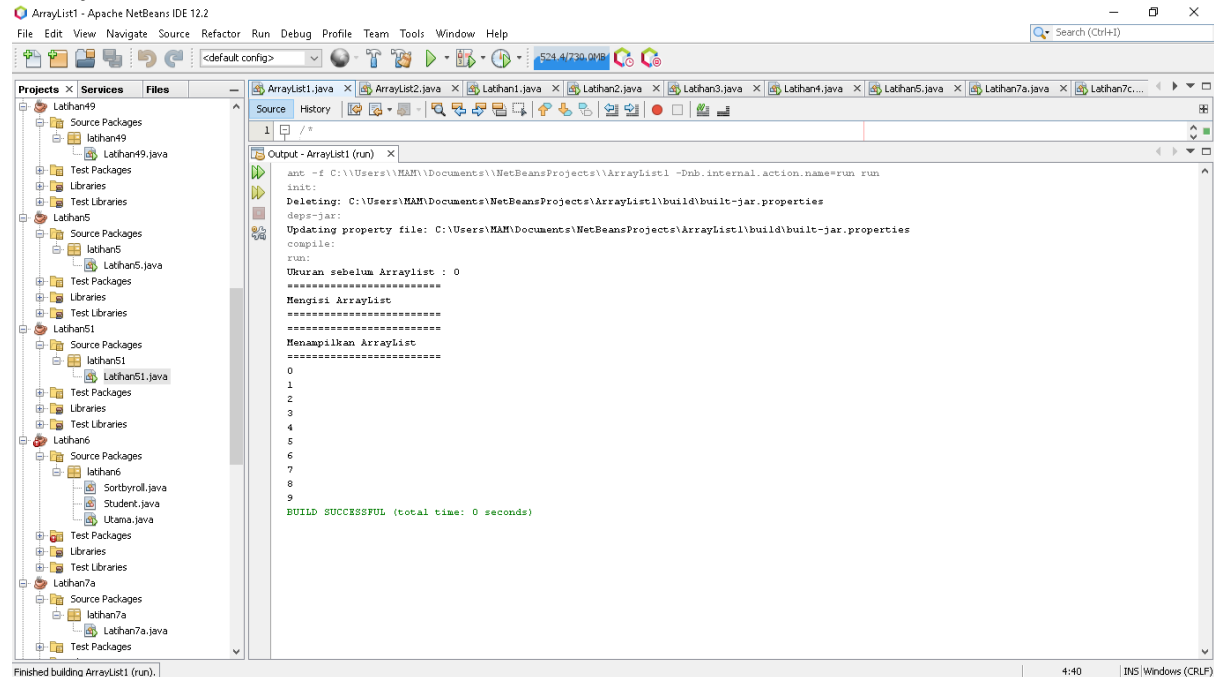
Metode pencarian yang ini adalah binary search, pada metode pencarian ini, data harus diurutkan terlebih dahulu. Pada metode pencarian ini, data dibagi menjadi dua bagian (secara logika), untuk setiap tahap pencarian. Data diambil dari posisi 1 sampai posisi akhir N . Kemudian cari posisi data tengah dengan rumus:  $(\text{posisi awal} + \text{posisi akhir}) / 2$ , Kemudian data yang dicari dibandingkan dengan data yang di tengah, apakah sama atau lebih kecil, atau lebih besar? Jika lebih besar, maka proses pencarian dicari dengan posisi awal adalah posisi tengah + 1 . Jika lebih kecil, maka proses pencarian dicari dengan posisi akhir adalah posisi tengah - 1. Jika data sama, berarti ketemu.

## Latihan 8c



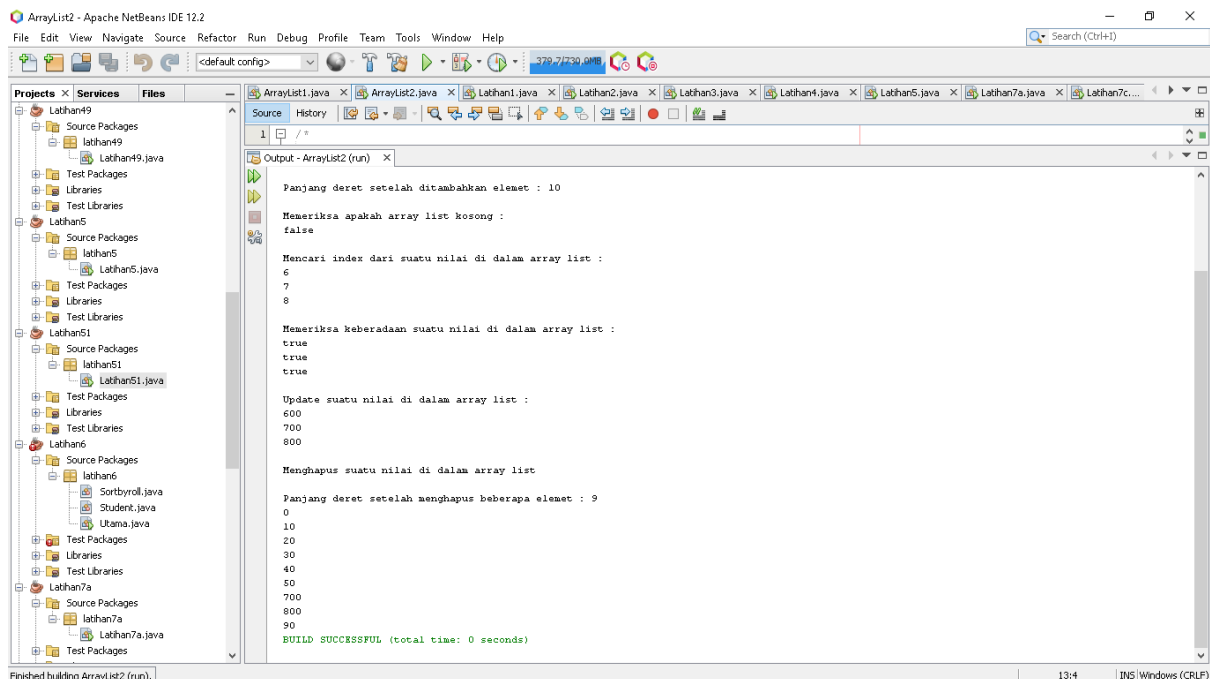
Mencari array yang ditentukan dari tipe data yang diberikan untuk nilai yang ditentukan menggunakan algoritma pencarian biner. Array harus diurutkan (seperti dengan metode `Arrays.sort()`) sebelum melakukan panggilan ini. Jika tidak diurutkan, hasilnya tidak ditentukan. Jika array berisi beberapa elemen dengan nilai yang ditentukan, tidak ada jaminan elemen mana yang akan ditemukan. indeks kunci pencarian, jika berada dalam array; jika tidak, (- (titik penyisipan) - 1). Titik penyisipan didefinisikan sebagai titik di mana kunci akan disisipkan ke dalam larik: indeks elemen pertama lebih besar dari kunci, atau `a.length` jika semua elemen dalam larik kurang dari kunci yang ditentukan. Perhatikan bahwa ini menjamin bahwa nilai yang dikembalikan akan  $\geq 0$  jika dan hanya jika kunci ditemukan. Jika daftar masukan tidak diurutkan, hasilnya tidak terdefinisi. Jika ada duplikat, tidak ada jaminan yang akan ditemukan. `Arrays.binarysearch()` bekerja untuk array yang juga bisa dari tipe data primitif. `Collections.binarysearch()` berfungsi untuk objek Koleksi seperti `ArrayList` dan `LinkedList`. Ada varian dari metode ini di mana kita juga dapat menentukan kisaran array yang akan dicari. Kita akan membahasnya serta mencari dalam array Object di posting selanjutnya

# Arraylist 1



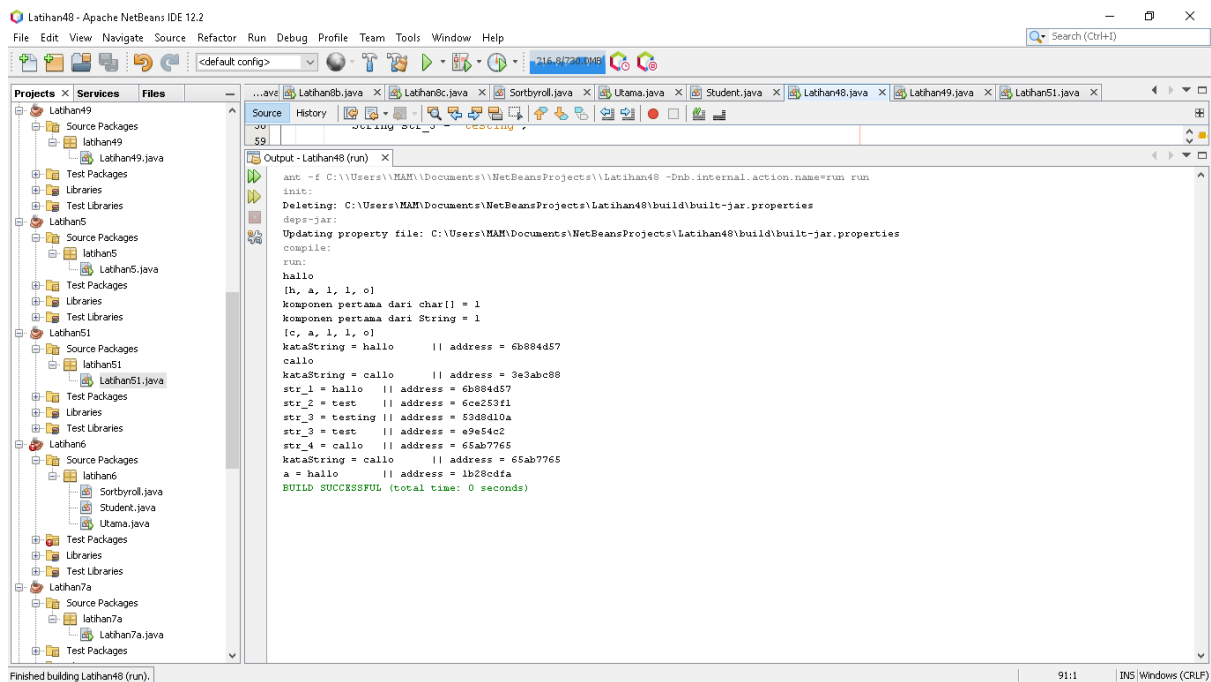
Arraylist memori = adalah bentuk deklarasi variable bertipe angka (integer) dengan nama memori. Lalu kita print untuk menampilkan ukuran sebelum arraylist dan memanggil nilai dari memori.size(). Lalu print mengisi arraylist. Kemudian lakukan loop for dengan 3statement. Statement pertama mendeklarasikan variabel int bernama I dan memberinya nilai 0, statemen kedua membandingkan nilai varial I dengan nilai 10. Jika nilai I kurang dari 10, maka loop for dijalankan sekali lagi. Statement ini dijalankan sebelum setiap pengulangan. Statement ketiga menambah nilai I hasil dari pengulangan ini akan dilakukan berulang 10 kali. Dan nilai I adalah kurang dari 10 (0 sampai 9). Selanjutnya dideklarasikan memori.add dengan nilai dari i. Lalu kita print untuk menampilkan ukuran sebelum ArrayList dan memanggil nilai dari memori.size(). Kemudian print mengisi ArrayList.

## Arraylist 2



. Mendeklarasikan sebuah array dengan tipe data int dengan variable nama deret “deret.add” untuk menambah element baru yang kita masukkan, kemudian menambahkannilai adanya “deret.size()” untuk mengetahui panjang dari length/ArrayList. Ada deret “deret.isEmpty()” untuk mereturn apakah ArrayList kosong atau tidak. “deret.indexOf(0)” ialah untuk mencari data dan mengetahui bahwa data yang dicari berada pada posisi index seberapa. Deret.contains()” untuk memeriksa apakah suatu nilai yang kita cari ada dalam ArrayList. Ada “dweret.set(index, input)” yang dimaksud disini yaitu index data yang ingin kita update dan input yang dimaksud adalah nilai atau data baru yang kita masukkan. “deret.get()” untuk mengambil data yang dicari itu isinya apa. Jika di “deret.remove()” maka akan terhapus data beserta indexnya. Sehingga datasetelahnya akan menempati index yang telah terhapus tadi. “for each java”, dengan data “deret” dikirim ke “nilai”, nanti dicetak.

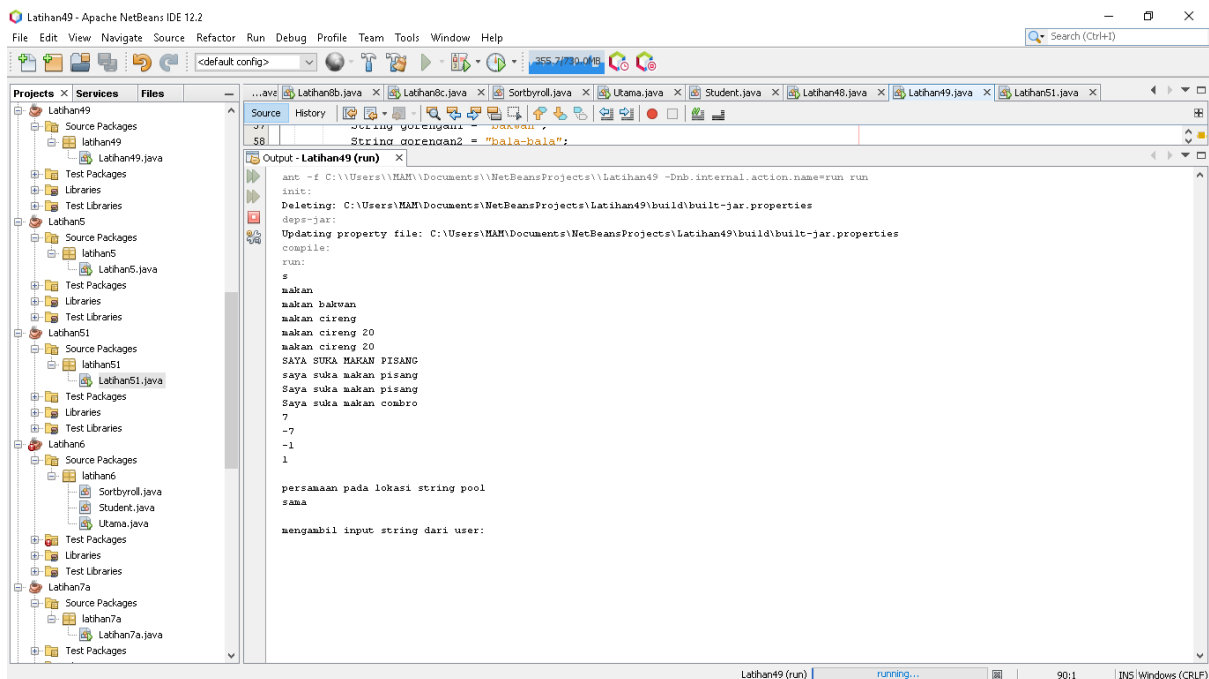
## Latihan 48



```
...ave Latihan48.java x Latihan49.java x Latihan50.java x Latihan51.java x
Source History
String str_3 = "testing";
59
Output - Latihan48 (run)
ant -f C:\Users\HAM\Documents\NetBeansProjects\Latihan48\build\build-jar.properties
init:
Deleting: C:\Users\HAM\Documents\NetBeansProjects\Latihan48\build\build-jar.properties
dope-jar:
Updating property file: C:\Users\HAM\Documents\NetBeansProjects\Latihan48\build\build-jar.properties
compile:
run:
hallo
[h, a, i, l, o]
komponen pertama dari char[] = 1
komponen pertama dari String = 1
[c, a, l, l, o]
kataString = hallo || address = 6b884d57
callo
kataString = callo || address = 3e3abc88
str_1 = hallo || address = 6b884d57
str_2 = test || address = 6ce253f1
str_3 = testing || address = 53d8d10a
str_3 = test || address = e9e54c2
str_4 = callo || address = 65ab7765
kataString = callo || address = 65ab7765
a = hallo || address = 1b28cdff
BUILD SUCCESSFUL (total time: 0 seconds)
```

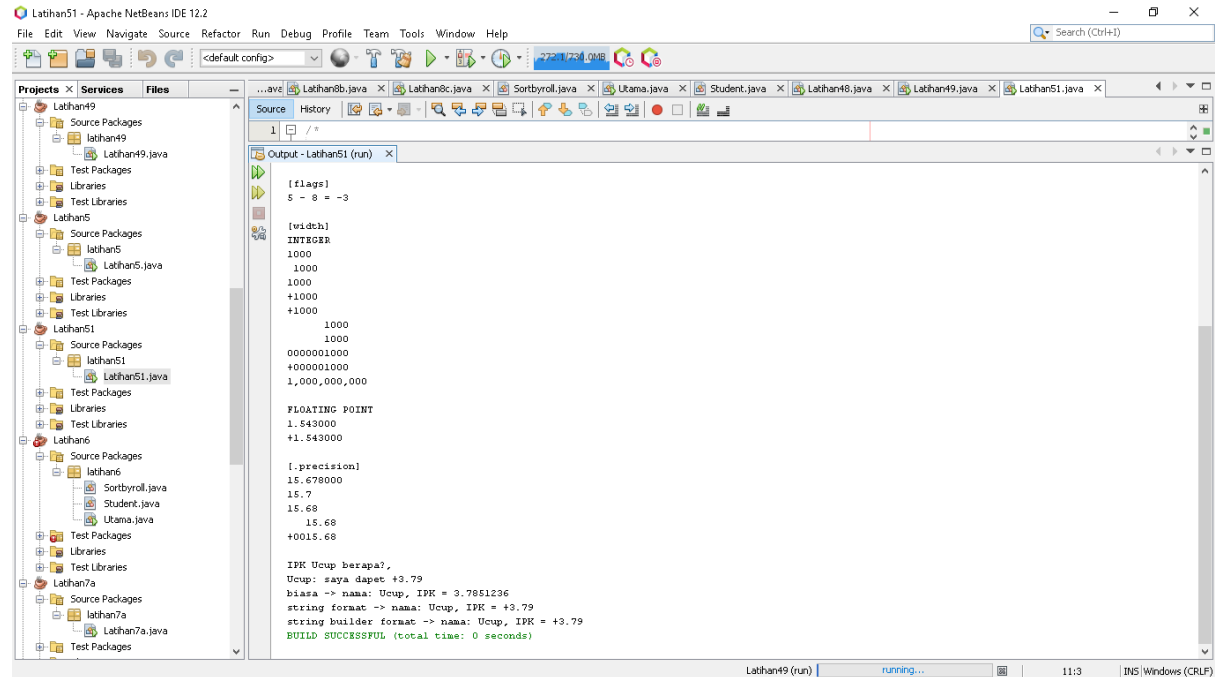
Pada implementasi dibawah ini sebenarnya adakah hanya menampilkan (diline 27 dan 28 ) apa yang diinputkan (pada line 21 dan 23), mengoutputkan (kataChar no (3)) yaitu “l” (pada line 32), Dan mengoutputkan kata string “l” pada line 33

## Latihan 49



Masukkan element string kemudian mengambil komponennya dengan “System.out.println(kalimat.charAt(5))”. Selanjutnya untuk operasi subString menggunakan “String kata = kalimat.substring(10,15)”. Kemudian penggabungan menggunakan String kalimat2 = kata + " bakwan" atayu cara yang kedua kata = kata + " cireng", namun cara kedua memiliki stringpool tersendiri. Dalam penggabungan dengan tipe data non string menggunakan “int jumlah = 20; String kalimat3 = kata + " " + jumlah”; jika ingin membuat string menjadi capslock(uppercase) maka menggunakan kalimat System.out.println(kalimat.toUpperCase()) namun jika ingin merubah kembali menjadi lowrcase maka menggunakan System.out.println(kalimat.toLowerCase()). Untuk penggantian komponen/replace, namun akan terjadinya pembentukan stringpool baru. Adanya compare/perbandingan “if” sebagai operasi persamaan, jika dua string dalam stringpool yang sama. Pengambilan inputan oleh user jika tidak ada dalam stringpool yang sama maka akan mencetak hasil tidak sama, tapi jika ingin mencetak hasil sama maka dengan “kataInput.equals(kataTest)” walaupun memorinya berbeda.

## Latihan 51



```
[flags]
5 - 8 = -3

[width]
INTEGER
1000
1000
1000
+1000
+1000

1000
1000
0000001000
+000001000
1,000,000,000

FLOATING POINT
1.543000
+1.543000

[.precision]
15.678000
15.7
15.68
15.68
+0015.68

IPK Ucup berapa?,
Ucup: saya dapet +3.79
biasa -> nama: Ucup, IPK = 3.7851236
string format -> nama: Ucup, IPK = +3.79
string builder format -> nama: Ucup, IPK = +3.79
BUILD SUCCESSFUL (total time: 0 seconds)
```

`System.out.println("Nama saya adalah " + nama + ", umur saya adalah " + umur);` perintah ini adalah perintah umum untuk mengeprint gabungan antara String dan integer. Hal ini bisa dilakukan dengan cara lainnya yakni dengan format string `System.out.printf("Nama saya adalah %s, umur saya adalah %d\n",nama,umur);` (perintah format string, menggunakan tanda “%” biasa disebut conversion). Tanda-tanda conversion adalah : f=floating point, d=integer, c=character, s=string, b=boolean struktur format = `%[argumen_index$][flags][width][.precision]conversion` ketika argumen nya itu yang dipakai sama maka dapat menggunakan argumen\_index seperti di bawah ini (tanda index adalah “\$”) `System.out.printf("%1$s, wahai %1$s, kemana saja kamu %1$s?\n",nama);` Flags menampilkan tanda “+” atau minus “-” untuk bilangan flags akan lebih terlihat jika ditambahi dengan width. Width untuk lebar variabel. Flags akan mengambil slot di dalam format. Flags kalo tanda “-” artinya rata kiri