

LAPORAN AKHIR
PEMBUATAN PROGRAM APLIKASI MENGHITUNG
LUAS BANGUN DATAR DAN VOLUME LIMAS



Disusun Oleh :
Khuszaimah Azizah (202013036)

Dosen Pengampu:
Slamet Triyanto, S.ST

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK KAMPAR
2021

KATA PENGANTAR

Puji dan syukur atas kehadiran Allah SWT, atas segala limpahan taufiq serta hidayah-Nya yang telah memberi penulis kesempatan untuk menyelesaikan Laporan Algoritma dan Pemograman ini. Adapun tujuan penulisan Laporan ini adalah untuk melengkapi Tugas ujian akhir semester Algoritma dan Pemograman.

Dalam proses pembuatan Laporan ini, tentunya penulis mendapat bimbingan, arahan, koreksi dan saran. Untuk itu penulis mengucapkan terima kasih kepada Bapak Slamet Triyanto,S.ST selaku dosen pengampu Algoritma dan Pemograman.

Penulis menyadari bahwa baik dari segi penulisan maupun isi, laporan ini masih memiliki kekurangan, oleh karena itu penulis sangat mengharapkan kritik yang membangun dan saran dari pembaca agar terbentuknya kesempurnaan laporan ini. Atas partisipasinya penulis mengucapkan terima kasih.

Bangkinang, Maret 2021

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL.....	v
BAB I	1
TINJAUAN PUSTAKA	1
A. Pengertian.....	1
B. Tujuan Praktikum.....	2
C. Alat dan Bahan.....	2
BAB II.....	3
PEMBAHASAN	3
A. Variabel Class/Static	3
B. Tipe Data Primitif	4
C. Literasi dalam Java.....	6
D. Oprator Aritmatika	6
E. Operator Relasional.....	7
F. Operator Logika	8
G. Modifier Akses.....	9
H. Perulangan Java.....	9
I. Kondisional If	11
J. Algoritma Sorting	14
BAB III	16
PEMBAHASAN	16
A. Langkah Kerja.....	16
B. Program Aplikasi Menghitung Luas Bangun Datar dan Volume Limas	17
BAB VI	27
PENUTUPAN	27
1. Kesimpulan	27
2. Saran.....	28
DAFTAR PUSTAKA.....	29

DAFTAR GAMBAR

Gambar 2.1 <i>variabel</i> kelas	3
Gambar 2.2 hasil <i>output</i>	4
Gambar 3.1 Aplikasi <i>Sublime</i>	16
Gambar 3.2 membuat <i>file</i> baru.....	16
Gambar 3.3 tampilan awal <i>new file</i>	17
Gambar 3.4 menampilkan <i>header file</i>	17
Gambar 3.5 menampilkan penamaan <i>class</i> dan <i>method</i> utama.....	17
Gambar 3.6 tampilan <i>source code login</i>	18
Gambar 3.7 Pemilihan.....	18
Gambar 3.8 pemilihan <i>menu</i>	19
Gambar 3.9 program aplikasi luas bangunan.....	19
Gambar 3.10 <i>Case 2</i>	20
Gambar 3.11 <i>Case 3</i>	21
Gambar 3.12 Aplikasi volume limas segi	21
Gambar 3.13 program pengurutan	22
Gambar 3.14 Membuka tampilan <i>cmd</i>	22
Gambar 3.15 memanggil <i>directory D</i>	23
Gambar 3.16 memanggil <i>file</i>	23
Gambar 3.17 membuka <i>file</i>	24
Gambar 3.18 <i>Program login</i>	24
Gambar 3.19 salah memasukan <i>username</i> dan <i>password</i>	24
Gambar 3.20 Pemilihan <i>menu</i>	25

Gambar 3.21 memilih bangun datar yang akan dihitung	25
Gambar 3.22 menentukan berapa kali perhitungan.....	26
Gambar 3.23 menampilkan hasil perhitungan dan pengurutan.....	26

DAFTAR TABEL

Tabel 2.1 karakter <i>escape-sequence</i>	6
Tabel 2.2 <i>Operator</i> aritmatika.....	7
Tabel 2.3 <i>Operator Relasional</i>	8
Tabel 2.4 <i>Operator</i> Logika	9

BAB I

TINJAUAN PUSTAKA

A. Pengertian

1. Pengertian *Java*

Menurut (Putri Indriyani,2012) *Java* adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut defenisi dari Sun, *Java* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. *Java* lebih banyak disebut sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena *Java* lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi *Java* memiliki tiga komponen penting, yaitu :

- a) *Programming*
- b) *Language Sepcification*
- c) *Application-Programming Interface*
- d) *Virtual-Machine Specification*

Kelebihan *Java*:

- a) Berorientasi objek, Memudahkan untuk mendesign dan mengembangkan program dengan cepat dan teliti, sehingga mudah digunakan. Salah satu bahasa pemrograman yang berorientasi objek secara murni .
- b) Mirip C++, mempunyai sintaks yang mirip dengan bahasa pemrograman C++. Sehingga para pengguna C++ banyak yang hijrah menggunakan *Java*.
- c) *Multiplatform*, Dapat digunakan di banyak sistem operasi.

- d) Perkembangan yang luas, dari *game* sampai sistem operasi handphone menggunakan program *java*. Misalnya handphone Sonny Ericsson dan OperaMini 3 yang bertipe.jar (*Java Archive*).
- e) Mempunyai pengumpulan sampah otomatis.

Kekurangan Java :

- a) Proses *Compile*, mengharuskan pengguna mengcompile programnya sebelum dijalankan, berbeda dengan bahasa pemrograman python yang tidak perlu mengcompile terlebih dahulu.
- b) Penggunaan Memori yang besar, berbeda dengan bahasa pemrograman lain yang hanya membutuhkan memori sedikit.”

B. Tujuan Praktikum

- 1. Untuk memahami Bahasa *Java*
- 2. Untuk memahami tipe data, percabangan, perulangan dan pengurutan
- 3. Untuk mengetahui *source code* yang digunakan untuk membuat aplikasi menghitung luas bangun datar dan volume limas

C. Alat Dan Bahan

- 1. Alat
 - a) Laptop
 - b) Mouse
- 2. Bahan
 - a) Aplikasi Sublime

BAB II

PEMBAHASAN

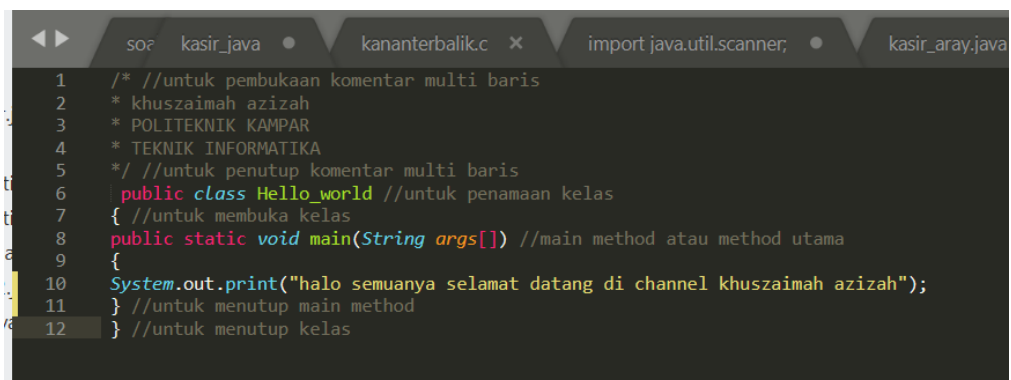
A. Variabel *Class/Static*

Variabel *class* atau biasa disebut *static* dideklarasikan dengan kata kunci *static* di dalam *class* tetapi diluar method, konstruktor, ataupun blok. Hanya ada satu salinan dari *variable class/static* ini per *class*.

Variabel *static* ini biasa nya di deklarasikan sebagai konstanta. Variabel konstanta nilainya tetap dan tidak berubah. Variabel *static* dibuat ketika program dimulai dan dihapus saat program berhenti. Nilai *default* nya sama dengan variabel *instance*.

Variabel *static* dapat diakses dengan memanggil nama *class*, misalnya Nama *Class*, NamaVariabel. Saat mendeklarasikan variabel *class* dengan *public static final*, maka nama variabel (konstanta) dituliskan semua dalam huruf kapital. Jika variabel *static* tidak dideklarasikan dengan *public final*, *syntax* penamaannya sama dengan variabel *instance* dan *local*.

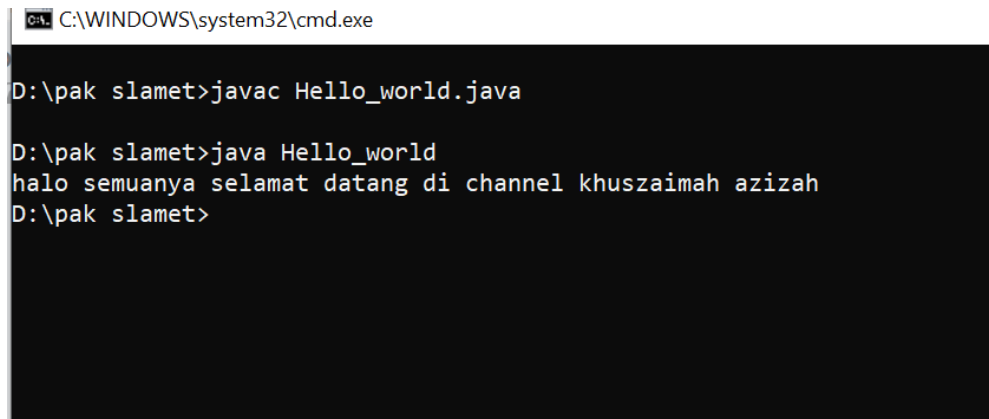
Berikut ini contoh penggunaan variabel kelas:



```
1  /* //untuk pembukaan komentar multi baris
2  * khuszaimah azizah
3  * POLITEKNIK KAMPAR
4  * TEKNIK INFORMATIKA
5  */ //untuk penutup komentar multi baris
6  public class Hello_world //untuk penamaan kelas
7  { //untuk membuka kelas
8  public static void main(String args[]) //main method atau method utama
9  {
10     System.out.print("halo semuanya selamat datang di channel khuszaimah azizah");
11 } //untuk menutup main method
12 } //untuk menutup kelas
```

Gambar 2.1 variabel kelas

Berikut adalah hasil output dari kode di atas :



```

C:\WINDOWS\system32\cmd.exe

D:\pak slamet>javac Hello_world.java

D:\pak slamet>java Hello_world
halo semuanya selamat datang di channel khuszaimah azizah
D:\pak slamet>
  
```

Gambar 2.2 hasil *output*

B. Tipe Data *Primitif*

Terdapat delapan tipe data *primitif* yang didukung dalam *java*. Tipe data *primitif* didefinisikan oleh bahasa dan diberi nama dengan sebuah kata kunci. Berikut ini tipe data primitif dalam *java*.

1. *Byte*

Tipe data *byte* merupakan *integer* 8-bit yang bersifat *signed*. Nilai *minimum* adalah -128, nilai *maksimum* adalah 127 dan nilai *default* adalah 0. Tipe data *byte* digunakan untuk menghemat ruang pada array yang besar. Contoh : *byte* a = 100, *byte* b = 50.

2. *Short*

Tipe data *short* merupakan *integer* 16-bit yang bersifat *signed*. Nilai *minimum* adalah -32,768, nilai *maksimum* adalah 32,767 dan nilai *default* adalah 0. Tipe data *short* juga dapat menghemat ruang seperti pada tipe data *byte*. Contoh: *short* s = 10000, *short* r = -20000.

3. *Int*

Tipe data *int* merupakan *integer* 32-bit yang bersifat *signed*. Nilai *minimum* adalah -2,147,483,648, nilai *maksimum* adalah 2,147,483,647 dan nilai *default* 0. Tipe data *int* digunakan secara

umum untuk menampung nilai-nilai integral. Contoh: `Int a = 100000`, `int b = -200000`.

4. *Long*

Tipe data *long* merupakan integer 64-bit yang bersifat signed. Nilai minimum adalah -9,223,372,036,854,775,808, nilai maksimum adalah 9,223,372,036,854,775,807 dan *default* adalah 0L. Tipe data *long* digunakan untuk menampung nilai-nilai dengan jangkauan nilai yang lebih dari tipe data *int*. Contoh: `int a = 100000L`, `int b = -200000L`.

5. *Float*

Tipe data *float* merupakan nilai *decimal* 32-bit presisi tunggal. *Float* biasanya digunakan untuk menghemat ruang pada *array* besar yang berisi nilai-nilai *decimal* dan nilai *default* adalah 0.0f. Tipe data *float* tidak pernah digunakan untuk nilai-nilai seperti nilai mata uang. Contoh: `float f1 = 234.5f`.

6. *Double*

Tipe data *double* merupakan nilai desimal 64-bit presisi ganda. *Double* biasanya digunakan sebagai tipe data *default* untuk nilai *decimal*. Nilai *default* adalah 0.0d. Tipe data *double* tidak pernah digunakan untuk nilai-nilai presisi seperti nilai mata uang. Contoh: `double d1 = 123.4`.

7. *Boolean*

Tipe data *Boolean* merepresentasikan satu bit informasi. Hanya ada dua nilai yang mungkin: *true* (benar) dan *false* (salah), tipe data ini digunakan sebagai penanda kondisi benar/salah. Nilai *default* adalah *false*. Contoh: `Boolean satu = true`.

8. *Char*

Tipe data *char* merupakan karakter *Unicode* 16-bit tunggal. Nilai minimum adalah ``\u0000`` dan nilai maksimum adalah ``\uffff``. Tipe data *char* digunakan untuk menampung karakter apa saja. Contoh: `char huruf A = `A``.

C. Literasi dalam *Java*

Bahasa Java mendukung beberapa *escape sequence* untuk literal *String* dan *char*.

Tabel 2.1 karakter *escape-sequence*

<code>\n</code>	Fungsi CR/LF (<i>Carriage Return and Line Feed</i>).Cursor pindah ke kolom pertama baris berikutnya.
<code>\b</code>	<i>Back Space, Cursor</i> mundur satu langkah pada baris yang sama.
<code>\r</code>	<i>Awal Baru</i>
<code>\t</code>	<i>Tab.Cursor</i> pindah ke posisi tabulasi berikutnya.
<code>\f</code>	Halaman Baru
<code>\'</code>	Untuk menampilkan karakter tanda petik tunggal(<i>single quotation mark</i>)
<code>\"</code>	Untuk menampilkan karakter tanda petik ganda(<i>double quotation mark</i>).

D. Operator Aritmatika

Operator aritmatika digunakan pada ekspresi matematik seperti pada operasi aljabar. Diansumsi kan bahwa variabel A bernilai 10 dan B bernilai 20, maka berikut ini daftar operator aritmatika:

Tabel 2.2 Operator aritmatika

Operator	Deskripsi	Contoh
+	Operator penjumlahan. Berfungsi untuk menambahkan operan	A + B hasilnya 30
-	Operator pengurangan. Berfungsi untuk mengurangi operan	A – B hasilnya - 10
*	Operator perkalian. Berfungsi untuk mengalikan operan.	A * B hasilnya 200
/	Operator pembagian. Berfungsi untuk membagi operan.	B/A hasilnya 2
%	Operator Modulus. Berfungsi mengurangi nilai operan di sebelah kiri dengan nilai operan di sebelah kanan dan mengembalikan sisa nilainya.	B%A hasilnya 0
++	Peningkatan(menambahkan nilai 1 operan)	B++ hasilnya 21
--	Penurunan (mengurangkan 1 nilai operan)	B—hasilnya 19

E. Operator Relasional

Operator ini digunakan untuk membandingkan dua operan yang umumnya dimanfaatkan dalam pembuatan struktur if. Berikut ini operator *relasional* yang didukung dalam java. Diasumsikan bahwa variabel A bernilai 10 dan B bernilai 20,maka :

Tabel 2.3 *Operator Relasional*

Operator	Deskripsi	Contoh
==	Memeriksa apakah nilai kedua operan sama atau tidak, jika sama maka kondisi bernilai benar.	(A==B) adalah tidak benar
!=	Memeriksa apakah nilai kedua operan sama atau tidak, jika tidak sama maka kondisi bernilai benar.	(A!=B) adalah benar
>	Memeriksa apakah nilai operan disebelah kiri lebih dari nilai operan disebelah kanan, jika ya maka kondisi bernilai benar.	(A>B) adalah tidak benar
<	Memeriksa apakah nilai operan disebelah kiri kurang dari nilai operan disebelah kanan, jika ya maka kondisi bernilai benar.	(A<B) adalah benar
>=	Memeriksa apakah nilai operan disebelah kiri lebih dari atau sama dengan nilai operan di sebelah kanan, jika ya maka kondisi bernilai benar	(A>=B) adalah tidak benar
<=	Memeriksa apakah nilai operan disebelah kiri kurang dari atau sama dengan nilai operan di sebelah kanan, jika ya maka kondisi bernilai benar	(A<=B) adalah benar

F. Operator Logika

Operator logika digunakan untuk mengetahui kondisi dari dua operan. Operator logika sering digunakan untuk membentuk struktur if. Berikut ini daftar operator logika. Diasumsikan bahwa nilai Boolean variabel A adalah *true* dan B adalah *false*, Maka :

Tabel 2.4 *Operator Logika*

Operator	Deskripsi	Contoh
&&	Operator logika <i>AND</i> . Jika kedua operator bukan nol, maka kondisi bernilai benar.	(A && B) adalah tidak benar.
	Operator logika <i>OR</i> . Jika ada operan yang bukan nol, maka kondisi bernilai benar.	(A B) adalah benar.
!	Operator logika <i>NOT</i> . Digunakan untuk membalik keadaan logika dari operan. Jika kondisi bernilai benar maka operan <i>NOT</i> akan membuatnya menjadi tidak benar.	!(A && B) adalah benar.

G. *Modifier* Akses

Java menyediakan beberapa *modifier* akses. Fungsinya untuk mengatur level akses untuk *class*, variabel, method, dan *constructor*. Berikut ini empat level akses:

1. *Default*: Dapat diakses oleh *package*, yang merupakan level akses *default*. Tidak memerlukan *modifier*.
2. *Private*: Hanya dapat diakses oleh *class*.
3. *Public*: Dapat diakses semua.
4. *Protected*: Dapat diakses oleh *package* dan semua *sub-class*.

Untuk menggunakan *modifier*, anda menambahkan kata kunci pada definisi *class*, method, atau variabel. *Modifier* berada di bagian awal statemen.

H. Perulangan *Java*

Perulangan digunakan untuk mengulang bagian tertentu dari struktur pemrograman sampai ditemukan kondisi dimana perulangan tersebut harus dihentikan. Bahasa pemrograman memiliki struktur pemrograman yang sama namun ditulis dengan *syntax* yang berbeda satu dnegna lainnya. Didalam bab

ini akan dibahas beberapa perulangan yang dapat digunakan dalam pemograman *java*.

1. Perulangan *While*

Perulangan *while* adalah struktur *control* yang memungkinkan anda mengualangi suatu proses dangan jumlah perulangan tertentu sampai ditemukan kondisi dimana perulangan akan dihentikan.Berikut ini *sintaks* dari perulangan *while*:

```
While(ekspresi_Boolean)
{
    //kode program yang akan diulang
}
```

Selama ekspresi_Boolean bernilai benar (*True*), maka statemen dalam struktur *while* akan terus dieksekusi.

2. Perulangan *do...while*

Perulangan *do...while* sama seperti perulangan *while*, tetapi perulangan *do...while* pasti akan dieksekusi minimal satu kali. Berikut sintaksnya:

```
Do
{
    //Statemen
}
While(ekspresi_boolean);
```

3. Perulangan *For*

Perulangan *for* adalah struktur *control repetitive* yang memungkinkan anda untuk menjalankan proses dengan jumlah perulangan tertentu(jumlah perulangan sudah diketahui sebelumnya). Berikut ini *sintaksnya*:


```

For (inisialisasi;ekspresi_boolean;update)
{
    //Statemen
}

```

Berikut ini aliran proses dalam perulangan *for*:

- a. Bagian inisialisasi dieksekusi pertama kali dan hanya sekali. Bagian ini memungkinkan anda untuk mendeklarasikan dan menginisialisasi variabel *control* perulangan.
- b. Setelah itu, ekspresi *Boolean* dievaluasi. Jika bernilai benar, statemen dalam struktur *for* akan dieksekusi. Jika tidak, statemen tidak dieksekusi dan proses berlanjut pada bagian setelah perulangan *for*.
- c. Setelah statemen dalam *for* dieksekusi, aliran proses kembali pada bagian *update*. Statemen *update* ini memungkinkan anda untuk meng-update nilai di dalam variabel *control* dalam *for*.
- d. Ekspresi *Boolean* kemudian dievaluasi lagi. Jika benar, statemen dalam perulangan *for* kembali dieksekusi, dan dilanjutkan ke bagian *update*. Jika tidak, perulangan *for* akan selesai dan proses berlanjut pada bagian setelah perulangan *for*.
- e. Ekspresi *Boolean* kemudian dievaluasi lagi. Jika benar, statemen dalam perulangan *for* kembali dieksekusi, dan dilanjutkan ke bagian *update*. Jika tidak, perulangan *for* akan selesai dan proses berlanjut pada bagian setelah perulangan *for*.

I. Kondisional *If*

Statemen *if* digunakan untuk membuat percabangan dari struktur program *java*. Dengan demikian, Anda bisa membuat beberapa struktur program yang akan dijalankan salah satunya, atau beberapa diantaranya,

apabila ada kondisi yang terpenuhi. Struktur *if* ini dikenali di seluruh bahasa pemrograman dan secara umum, cara penggunaannya tidak berbeda jauh.

1. Statemen *if*

Statemen *if* dalam *java* terdiri dari sebuah ekspresi *Boolean* yang diikuti dengan satu statemen atau lebih. Berikut ini sintaks untuk statemen *if*:

```
If(ekspresi_boolean)
{
    //Statemen akan dieksekusi jika ekspresi Boolean
    //bernilai benar
}
```

2. Statemen *if...else*

Statemen *if* dapat diikuti dengan statemen *else* yang opsional agar anda bisa menyusun struktur program yang akan dijalankan apabila ekspresi *Boolean* yang diuji menghasilkan nilai salah. Berikut contoh sintaksnya:

```
If(ekspresi_boolean)
{
    //Dieksekusi jika ekspresi Boolean bernilai benar
}
else
{
    //Dieksekusi jika ekspresi Boolean bernilai salah
}
```

3. Statemen *if...else if...else*

Statemen *if* dapat diikuti dengan statemen *else if...else opsional*. Saat menggunakan statemen *if*, *else if*, *else* ada beberapa hal yang harus anda perhatikan:

- a. Sebuah statemen *if* dapat memiliki nol atau satu statemen *else* dan harus digunakan setelah statemen *else if*.

- b. Sebuah statemen *if* dapat memiliki nol atau banyak statemen *else if* dan harus digunakan sebelum statemen *else*.
- c. Setelah sebuah statemen *else if* dieksekusi, statemen *else if* atau *else* yang lain tidak akan diperiksa(akan langsung melompat pada kode setelah *else* terakhir)

4. Statemen *Switch*

Statemen *Switch* memungkinkan sebuah variabel diperiksa kesamaannya dengan daftar nilai-nilai yang ada. Setiap nilai dalam daftar disebut *case*. Berikut ini sintaks statemen *switch*:

```
Switch(expression)
{
case nilai:
//Statemen
break;//opsional
case nilai:
//Statemen
break; //opsional
// anda dapat menggunakan beberapa statemen case
default: //optional
//Statements
}
```

Berikut ini aturan penggunaan statemen *switch*:

Variabel yang digunakan dalam *switch* hanya boleh berupa *byte*, *short*, *int*, atau *char*. Anda dapat menggunakan beberapa *case* didalam *switch*. Setiap *case* diikuti dengan nilai yang akan dibandingkan dan tanda titik dua(:). Nilai *case* harus sama tipe datanya dengan variabel dalam *switch* dan harus berupa konstanta atau *literal*. Ketika variabel yang dibandingkan sama dengan dengan *case*, statemen pada *case* tersebut akan dieksekusi sampai pada kata kunci *break*. Ketika sampai pada kata kunci

break, *switch* berhenti dieksekusi, dan aliran proses melompat pada kode sesudah statemen *switch*.

Tidak semua *case* memerlukan *break*. Jika tidak ada *break*, aliran proses akan terus dijalankan pada *case* selanjutnya sampai pada *break*. Statemen *switch* dapat memiliki *case default*, yang diletakan pada akhir statemen. *Default* dijalankan jika tidak ada *case* yang dijalankan.

J. Algoritma *Sorting*

1. *Insertion Sorting*

Insertion Sorting adalah algoritma pengurutan sederhana yang *relative efisien* untuk *list* data kecil dan sering digunakan sebagai bagian dari algoritma yang lebih digunakan sebagai bagian dari algoritma yang lebih mutakhir. Algoritma ini bekerja dengan cara mengambilelemen dari *list* secara satu per satu dan menyisipkan elemen tersebut ke dalam posisi yang benar di *list* baru yang telah diurutkan (N. Wirth 1985).

2. *Selection Sort*

Algoritma *Selection Sort* memilih elemen elemen maksimum/minimum *array*, lalu menempatkan elemen maksimum/minimum itu pada awal atau akhir *array* (tergantung pada urutannya *ascending/descending*).Selanjutnya elemen tersebut tidak disertakan pada proses selanjutnya. Karena setiap kali *Selection Sort* harus membandingkan elemen – elemen data, algoritma ini termasuk dalam *Comparison Based Sorting* (Ananda,et al.2009).

Terdapat dua pendekatan dalam metode pengurutan dengan *Selection Sort*

- a. Algoritma pengurutan maksimum(*Maximum Selection Sort*), yaitu memilih elemen maksimum sebagai basis pengurutan.

- b. Algoritma pengurutan minimum(*Minimum Selection Sort*), yaitu memilih elemen minimum sebagai basis pengurutan.

3. *Quick Sort*

Quick Sort adalah salah satu jenis algoritma jenis pengurutan dengan konsep divide and conquer dengan mengandalkan operasi partisi, untuk mempartisi sebuah elemen array yang dipilih yang disebut sebagai *pivot* (Cormen. At al.2009).

4. *Merge Sort*

Algoritma *Merge Sort* mengambil keuntungan dari kemudahan menggabungkan daftar yang sudah diurutkan ke dalam daftar urutan yang baru. Dimulai dengan proses membandingkan setiap dua elemen(misalnya, 1 dengan 2,3 dengan 4, dan seterusnya) dan menukarnya jika nilai pertama lebih besar. Kemudian menggabungkan masing-masing dua elemen tersebut menjadi empat elemen dan seterusnya.(N.Wirth 1985).

5. *Array.sort() Java*

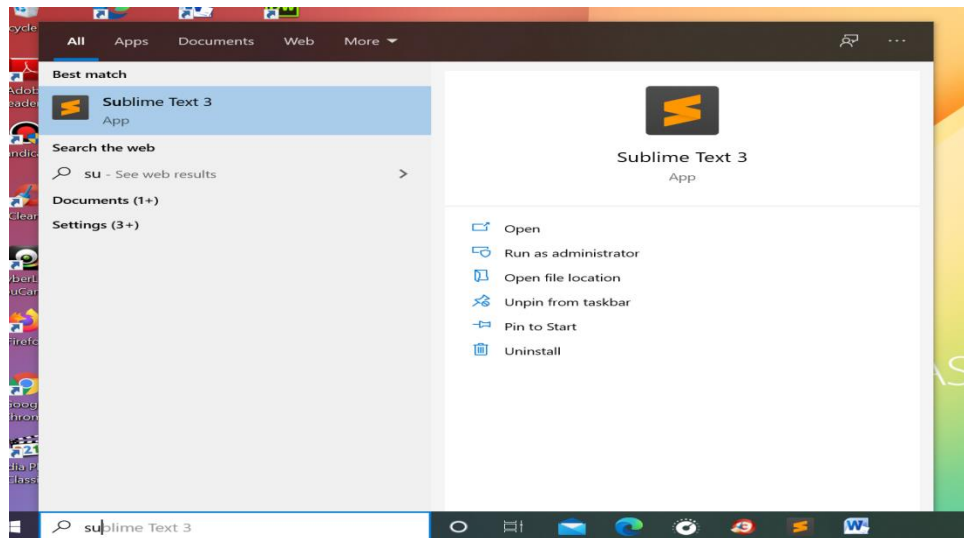
Di dalam *class library Java* terdapat sebuah method yang digunakan untuk mengurutkan data. Method tersebut berada di dalam *class Array*.

BAB III

PEMBAHASAN

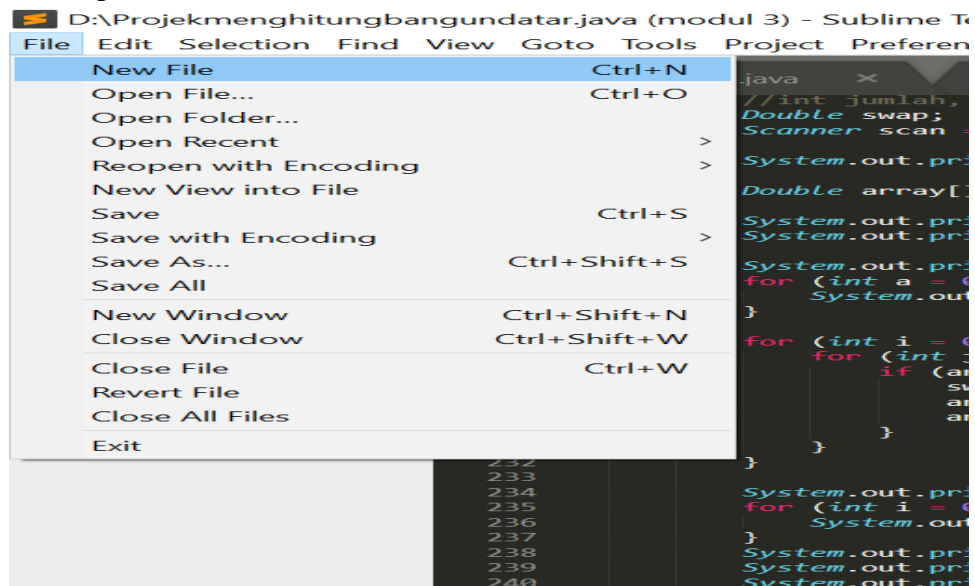
A. Langkah Kerja

1. Pastikan dilaptop/*computer* anda sudah tersedia aplikasi sublime. Untuk membuat *Source Code*.



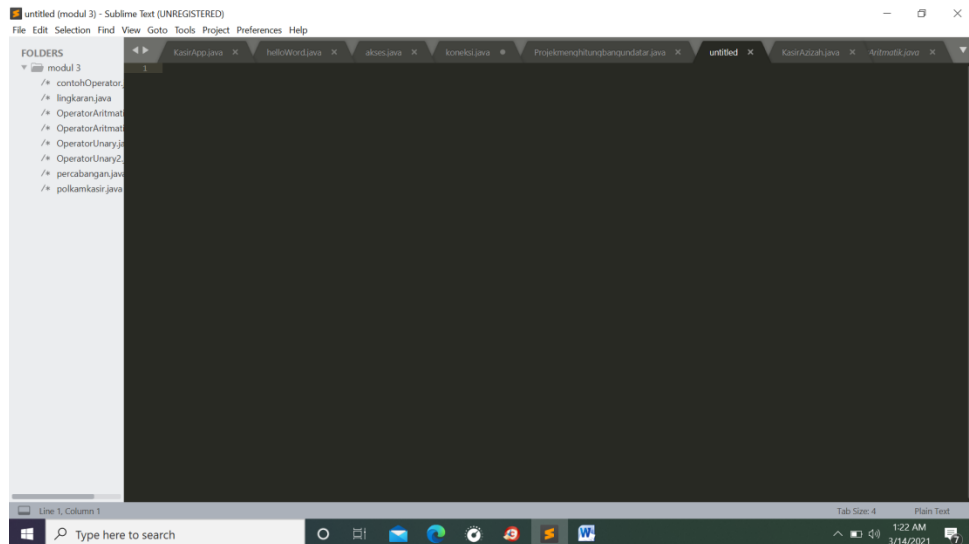
Gambar 3.1 Aplikasi *Sublime*

2. Buka aplikasi Pilih File > New File.



Gambar 3.2 membuat *file* baru

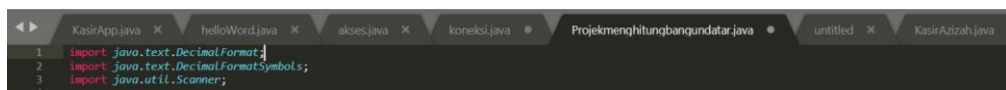
3. Maka akan muncul tampilan kosong seperti ini.



Gambar 3.3 tampilan awal *new file*

4. Kemudian buat *Source Code* mengenai perhitungan luas bangun datar dan volume limas

B. Program Aplikasi Menghitung Luas Bangun Datar dan Volume Limas



Gambar 3.4 menampilkan *header file*

Pada baris ke-1 sampai baris ke-3, berisi *headerfile* dimana baris ke-1 untuk bilangan berkoma, barisan ke-2 sama saja dengan baris pertama tetapi disini menggunakan *symbols* untuk memunculkan rupiah(Rp) dan baris ke-3 *header* untuk melakukan *Scanner*.



Gambar 3.5 menampilkan penamaan *class* dan *method* utama

Pada baris ke-5 merupakan penamaan *class* dimana penamaan ini sangat penting karena nama *class* harus sama dengan file yang disimpan dengan indeks *.java*. Selanjutnya baris ke-7 yang merupakan method utama dan pada baris ke-9 ada tanda tutup kurung kurawal untuk menutup perintah.

```

11  static void login() {
12      String data[] = {"Azizah", "202013036"};
13      String nama, sandi;
14      System.out.println("Selamat Datang di");
15      System.out.println("POLITEKNIK KAMPAR");
16      System.out.println("AA");
17      System.out.println("A A");
18      System.out.println("AA AA");
19      System.out.println("AAA AAA");
20      System.out.println("AAAAAAAAAAAA");
21      System.out.println("AAA AAA");
22      System.out.println("AAA AAA");
23      System.out.println("AAA AAA");
24      System.out.println("AAA AAA");
25      System.out.println("-Teknik Informatika-");
26      System.out.println("Bahasa Java");
27      Scanner apa = new Scanner(System.in);
28      System.out.print("USERNAME : ");
29      nama = apa.nextLine();
30      System.out.print("PASSWORD : ");
31      sandi = apa.nextLine();
32      System.out.println();
33  }

```

Gambar 3.6 tampilan *source code login*

Baris ke-11 merupakan fungsi *void* penamaan kelas untuk *login* ke program. Pada baris ke-12 merupakan *array* yang akan dimasukan untuk login. Baris ke-13 untuk mendeklarasikan tipe data *string*. *Login* ini digunakan untuk masuk ke aplikasimenghitung luas bangun datar dan volume limas. Sedangkan baris ke-14 sampai baris ke-25 digunakan untuk menampilkan inputan dilayar berupa keterangan nama politeknik dan prodi dengan perintah `string.out.println`. Lanjut pada baris ke-29 dan baris ke-31 digunakan untuk menginputkan *array* yang sudah ditanamkan.

```

37  if (nama.equals(data[0]) && sandi.equals(data[1])) {
38      System.out.println("t-----");
39      System.out.println("\t\tHallo Guys....");
40      System.out.println("t-----");
41      System.out.println(menu());
42  } else {
43      System.out.println("Sorry username is not available");
44      login();
45  }
46  }
47  }
48

```

Gambar 3.7 Pemilihan

Pada baris-37 merupakan pemilihan apabila benar maka akan melaksanakan program *if*, jika program berhasil maka baris ke-38 sampai baris ke-40 akan muncul dilayar. Namun, jika pernyataan atau *password* salah maka baris ke-44 yang akan muncul kelayar.

Disini menggunakan *static void bangun()* { pada baris ke-75 untuk penamaan *class* memilih luas bangunan. Selanjutnya, baris ke-81 merupakan pendeklarasian tipe data *integer*. Pada baris ke-84 adalah deklarasi array menggunakan tipe data *string*. Sedangkan baris ke-85 sampai baris ke-87 adalah *indeks array* dari 0 sampai 2. Lanjut ke percabangan *switch case* yang dimulai pada baris ke-93 yaitu *switch (listterima)* { dan *case 1* pada baris ke-94 dimana kita akan menghitung luas lingkaran. Dalam *case 1* juga terdapat perulangan pada baris ke-102 dimana *while (0 < i)* { dibaca jika *i* kurang dari 0 maka program akan berjalan, program berhenti jika 0 kurang 1.

Selanjutnya kita melakukan pengurutan dengan cara memanggil *class urut(penampung)*; yang terjadi pada baris ke-114, lalu pada baris ke-115 ada *menu()*; yang digunakan untuk pindah ke fungsi *void main menu*. Baris ke-117 ada *break*; untuk mengakhiri dari *case 1*.

```

118      case 2:
119      |
120      System.out.println("Menghitung Luas persegi");
121      System.out.println("-----");
122      System.out.print("Berapa kali anda ingin menghitung : ");
123      i = input.nextInt();
124      Double penampung2[] = new Double[i];
125
126      while(0 < i) {
127          System.out.print("Masukan Panjang=>");
128          Double lebar, luas2, panjang;
129          Scanner terimal = new Scanner(System.in);
130          panjang = terimal.nextDouble();
131          System.out.print("Masukan lebar=>");
132          lebar = terimal.nextDouble();
133          luas2 = lebar * panjang;
134          penampung2[i-1] = luas2;
135          System.out.println("Luas Lingkaran : " + luas2);
136          System.out.println("-----");
137          System.out.println("Proses Melakukan Perhitungan Telah Selesai");
138          i--;
139      }
140      urut(penampung2);
141      menu();
142
143      break;

```

Gambar 3.10 Case 2

Alur programnya sama dengan *case 1* namun yang membedakan di *case 2* ini kita akan menghitung luas persegi. Pada baris ke-124 berisikan pendeklarasian tipe data *double* dan pada baris ke-133 kita melakukan perhitungan dengan menggunakan operator aritmatika berupa kali (*). Dan di akhiri dengan *break*; baris ke-143 untuk mengakhiri dari *case 2*.

```

144         case 3:
145             System.out.print("Proses berakhir");
146             System.exit(0);
147
148         default:
149             //jika mSasukan salah
150             System.out.print("Kode tidak tersedia" + listterima + "System akan keluar");
151             System.exit(0);
152             System.out.println("1. Kembali ke Menu");
153             System.out.println("2. Keluar");
154             Scanner apa = new Scanner(System.in);
155             int kondisi;
156             System.out.print("Apakan Anda ingin melanjutkan program : ");
157             kondisi = apa.nextInt();
158             if (kondisi == 1) {
159                 menu();
160             } else {
161                 System.exit(0);
162             }
163     }
164 }
165
166

```

Gambar 3.11 Case 3

Disini kita menampilkan case 3 dimana jika proses berjalan sesuai dengan prosedur maka akan muncul di layar Proses berakhir seperti baris ke-146. Namun jika tidak ada pemilihan yang benar maka program akan keluar otomatis karna menggunakan default; baris ke-149 . Pada baris ke-159 dan ke-161 terdapat *if* dan *else* . Apabila user memilih *if* maka akan kembali ke menu, namun jika *user* memilih *else* maka program otomatis akan keluar.

```

169 static void volume() {
170     Double volume, p, l, t;
171     Scanner oke = new Scanner(System.in);
172     System.out.println();
173     System.out.println("-----");
174     System.out.println("          Volume Limas Segi");
175     System.out.println("-----");
176
177     System.out.println();
178     System.out.print("Masukan panjang limas : ");
179     p = oke.nextDouble();
180     System.out.print("Masukan lebar limas : ");
181     l = oke.nextDouble();
182     System.out.print("Masukan Tinggi limas : ");
183     t = oke.nextDouble();
184     volume = p * l * t / 3;
185
186     System.out.println("Jadi Volume Limas Segi Empat Adalah: " + volume);
187     System.out.println();
188     System.out.println("1. Kembali ke Menu");
189     System.out.println("2. Keluar");
190     Scanner apa = new Scanner(System.in);
191     int kondisi;
192     System.out.print("Apakan Anda masih ingin melanjutkan program lain : ");
193     kondisi = apa.nextInt();
194     if (kondisi == 1) {
195         menu();
196     } else {
197         System.exit(0);
198     }
199 }
200
201
202
203
204

```

Gambar 3.12 Aplikasi volume limas segi

Pada pemograman kali ini kita tidak menggunakan *bubble sort* atau pengurutan, agar kita mengetahui apa sih bedanya jika kita menggunakan *bubble sort* dan tidak. Baris ke-169 merupakan fungsi *void* penamaan kelas untuk menghitung volume limas. Pada baris ke-170 berisi pendeklarasian variabel tipe data *double* ,Baris ke-184 terjadi perhitungan dengan menggunakan operator aritmatika yaitu kali(*) dan bagi (/) dan pada baris ke-

192 terdapat pendeklarasian variabel tipe data integer. Disini kita juga menggunakan *if else* pada baris ke-195 dan baris ke-197.

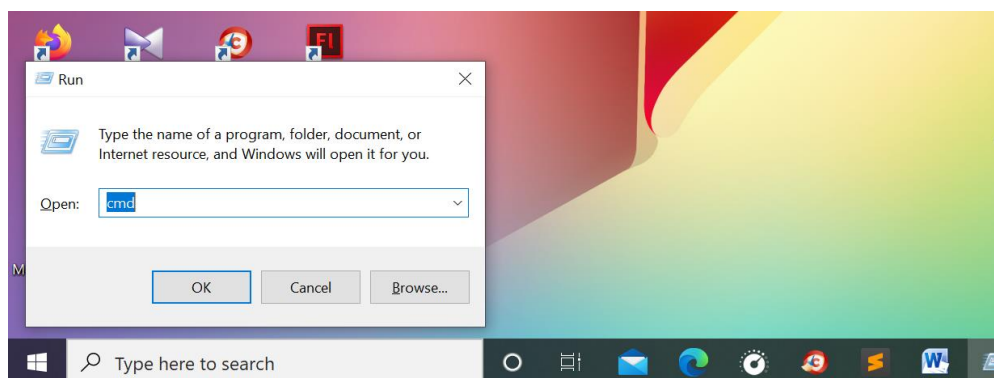
```

208 static void urut(Double value[]) {
209     //Jenis jumlah, 1, 2, swap;
210     Double swap;
211     Scanner scan = new Scanner(System.in);
212     System.out.print("Masukkan Jumlah Angka Yang ingin anda urutkan :");
213
214     Double array[] = value;
215
216     System.out.println("\nMasukkan " + value.length + " Buah Barang");
217     System.out.println("=====");
218
219     System.out.println("\nAngka Belum Terurut");
220     for (int a = 0; a < value.length; a++) {
221         System.out.print(array[a] + " | ");
222     }
223
224     for (int i = 0; i < (value.length - 1); i++) {
225         for (int j = 0; j < value.length - i - 1; j++) {
226             if (array[j] > array[j + 1]) {
227                 swap = array[j];
228                 array[j] = array[j + 1];
229                 array[j + 1] = swap;
230             }
231         }
232     }
233
234     System.out.println("\nAngka yang sudah Terurut adalah :");
235     for (int i = 0; i < value.length; i++) {
236         System.out.print(array[i] + " | ");
237     }
238     System.out.println();
239     System.out.println("1. Kembali ke Menu");
240     System.out.println("2. Keluar");
241     Scanner apa = new Scanner(System.in);
242     int kondisi;
243     System.out.print("Apakan Anda masih ingin program lain : ");
244     kondisi = apa.nextInt();
245     if (kondisi == 1) {
246         menu();
247     } else {
248         System.exit(0);
249     }
250 }
251 }

```

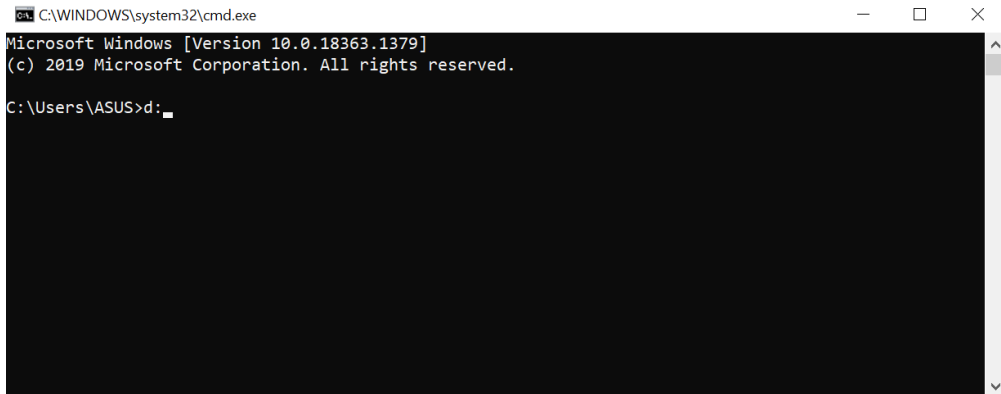
Gambar 3.13 program pengurutan

Disini kita membuat program pengurutan, dimana kita akan mengurutkan hasil dari perhitungan luas bangun datar. Pada baris ke-208 merupakan fungsi untuk mengurutkan angka. Baris ke-210 merupakan deklarasi tipe data double ,dimanaswap artinya menukar. Pada baris ke-215 menjelaskan bahwa tipe data array berupa *double*. Pada baris ke-249 terdapat tanda tutup kurung kurawal (}) untuk menutup fungsi dari *void* dari pengurutan, tutup kurung kurawal (}) pkedua baris ke-250 untuk menutup method utama dan tutup kurung kurawal (}) pkedua baris ke-251 untuk menutup nama *class* atau kelas.



Gambar 3.14 Membuka tampilan cmd

Langkah berikutnya kita masuk ke cmd untuk melihat hasil program. Ketik windows+R > ketik cmd > enter. Maka, akan tampil seperti gambar dibawah samping sebelah kanan.



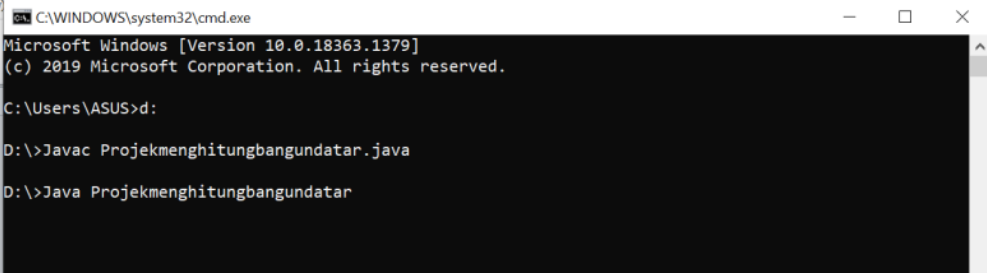
Gambar 3.15 memanggil *directory D*

Ketikan nama *directory* tempat menyimpan file program mesin kasir sederhana ketik d: > enter.



Gambar 3.16 memanggil *file*

Selanjutnya panggil file Javac (nama file).java> enter



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

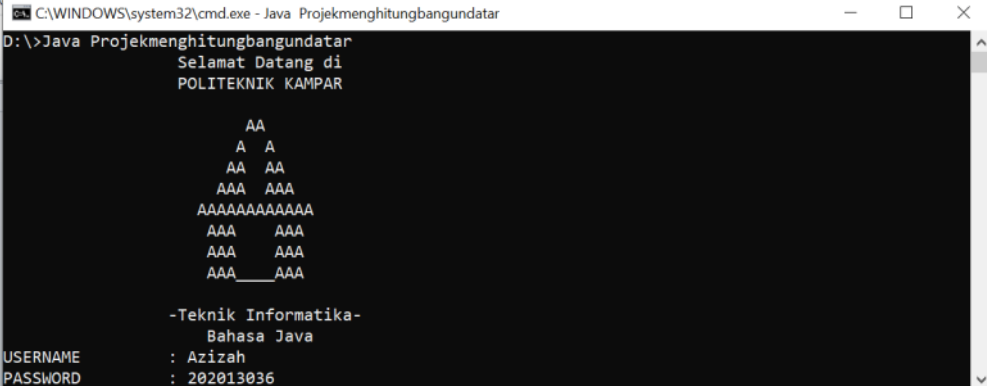
C:\Users\ASUS>d:

D:\>Javac Projekmenghitungbangundatar.java

D:\>Java Projekmenghitungbangundatar
  
```

Gambar 3.17 membuka *file*

Selanjutnya ketik *Java* (nama file)>enter



```

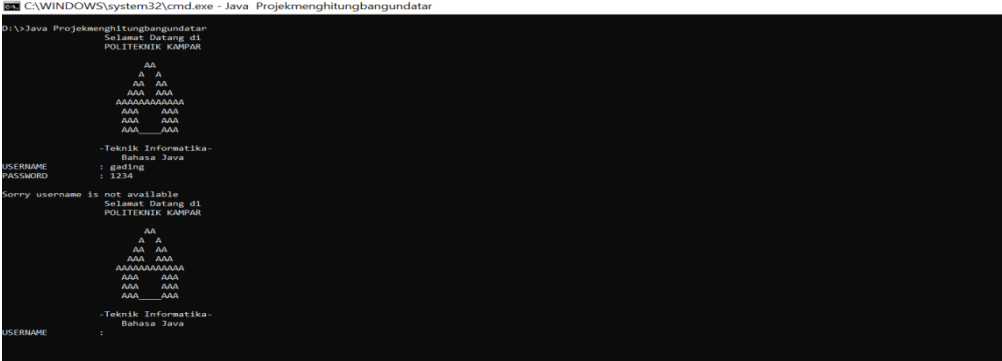
C:\WINDOWS\system32\cmd.exe - Java Projekmenghitungbangundatar
D:\>Java Projekmenghitungbangundatar
Selamat Datang di
POLITEKNIK KAMPAR

      AA
     A A
    AA AA
   AAA AAA
  AAAAAAAAAA
 AAA  AAA
 AAA  AAA
 AAA  AAA

-Teknik Informatika-
 Bahasa Java
USERNAME : Azizah
PASSWORD : 202013036
  
```

Gambar 3.18 *Program login*

Tampilan diatas adalah tampilan selamat datang dan *login* ke program aplikasi perhitungan luas bangun datar dan volume limas. Di sini *user* diminta untuk memasukan *user* dan *password* sesuai dengan ketentuan. Apabila salah dalam memasukan *user name* atau *password* maka tampilan seperti dibawah ini.



```

C:\WINDOWS\system32\cmd.exe - Java Projekmenghitungbangundatar
D:\>Java Projekmenghitungbangundatar
Selamat Datang di
POLITEKNIK KAMPAR

      AA
     A A
    AA AA
   AAA AAA
  AAAAAAAAAA
 AAA  AAA
 AAA  AAA
 AAA  AAA

-Teknik Informatika-
 Bahasa Java
USERNAME : gading
PASSWORD : 1234

Sorry username is not available
Selamat Datang di
POLITEKNIK KAMPAR

      AA
     A A
    AA AA
   AAA AAA
  AAAAAAAAAA
 AAA  AAA
 AAA  AAA
 AAA  AAA

-Teknik Informatika-
 Bahasa Java
USERNAME :
  
```

Gambar 3.19 salah memasukan *username* dan *password*

BAB VI

PENUTUPAN

1. Kesimpulan

Berdasarkan tahapan yang telah dilakukan dalam perencanaan program Aplikasi Menghitung Luas Bangun Datar dan Volume Limas, maka penulis menyimpulkan beberapa hal sebagai berikut :

- a. Dalam pemograman aplikasi menghitung luas bangun datar dan volume limas ini telah berhasil untuk menyelesaikan tugas proyek ujian akhir semester 1 mata kuliah Algoritma dan Pemograman .
- b. Program ini menggunakan bahasa *Java* dan perintah berupa:
 1. Daftar *header file*
 2. Tipe data
 - a. *Integer*
 - b. *Double*
 3. Operator Aritmatika
 4. Perulangan *For*
 5. Pemilihan *Switch Case*
 6. Percabangan *if else*
 7. Variabel
 8. *Bubble Sort* (pengurutan)
- c. Program ini dapat mempermudah dalam menghitung program dalam satu perintah.
- d. Program ini menyediakan beberapa menu untuk mempermudah proses perhitungan
- e. Program ini terdiri atas prodesur yang didalam nya terdapat perintah pemilihan dan perulangan.

2. Saran

Saran yang dapat diberikan penulis untuk pengembangan program mesin kasir sederhana selanjutnya dimasa yang akan datang adalah :

- a. Pada program aplikasi menghitung luas bangun datar dan volume limas ini belum menggunakan data base sehingga program belum bisa menyimpan data dengan benar.
- b. Program ini juga dapat dibuat dengan tampilan yang lebih menarik.

DAFTAR PUSTAKA

AF Sallaby, F. U. (2015). Aplikasi Widget berbasis java. Jurnal Media Infotama.

Enterprise, J. (2019). Lancar Java dan Javascript. PT Elex Media Komputindo,
Jakarta: PT Elex Media Komputindo.

Maulana, M. R. (2017). Komparasi Algoritma Sorting pada Bahasa Pemograman
Java. STMIK Widya Pratama Pekalongan, 37.