



TUGAS BESAR I PENGENALAN KOMPUTASI

Kelompok 5 K-18



KU1102





/ANGGOTA



/01 19622094

> Renaldy Arief
Susanto

/02 19622144

> Melati Anggraini

/03 16522044

> Rafid Ahmad Rabbani

/04 16522074

> Kimi Rafif Asyadda





/TUJUAN



- Mahasiswa dapat menggunakan teknik-teknik *computational thinking* (*decomposition*, *abstraction*, *pattern recognition*, dan *algorithm design*) serta menggunakan teknik-teknik *algorithmic thinking* dalam suatu persoalan.
- Mahasiswa bekerja sama dan berkomunikasi dalam kelompok serta mempresentasikan hasil kerja kelompok.





/PENDAHULUAN

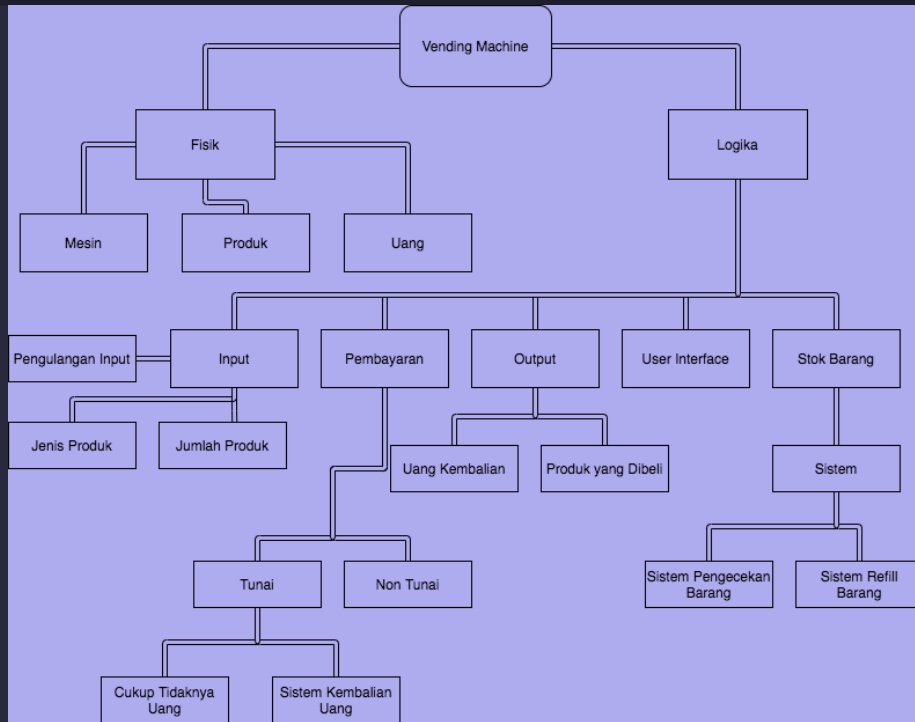


- Seperti yang kita tahu, pada tahap persiapan bersama Institut Teknologi Bandung kita diberikan mata kuliah pengenalan komputasi yang fungsinya melatih keterampilan dasar cara berfikir komputasi, algoritma, dan pemrograman. Diberikanlah tugas besar untuk mengaplikasikan ilmu-ilmu yang kita dapat.
- Kelompok kami memutuskan untuk membuat program vending machine karena bisa di-dekomposisi dengan cukup detail tanpa membuat kami kesulitan. Sebelum membuat kode dari program, kami perlu memecah masalah menjadi bagian yang lebih kecil dan membuat diagram alir.





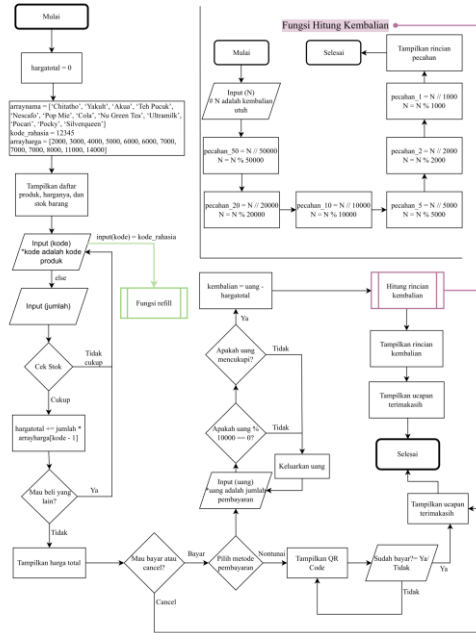
/DEKOMPOSISI MASALAH



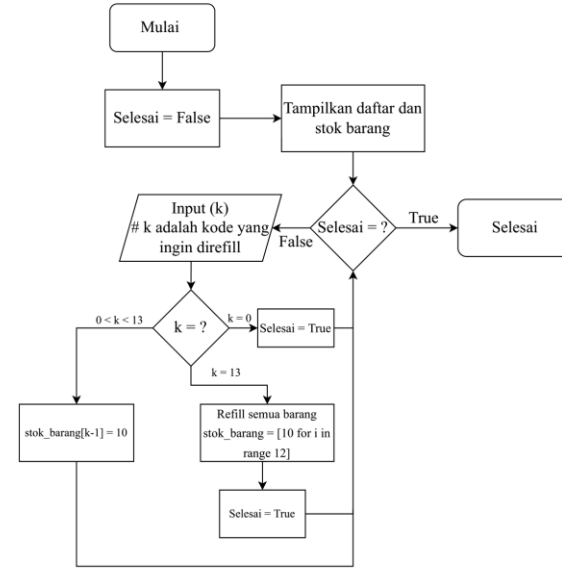


/DIAGRAM ALIR

Diagram Alir Vending Machine Kelompok 5



Fungsi refill barang





/ALUR KODE



INPUT

Masukkan dari
user.



PROSES

Pengolahan data
dari masukan user.



OUTPUT

Hasil / keluaran
dari data yang
sudah melalui
pemrosesan.





/ALUR KODE



INPUT

- Input kode barang
- Input kode refill
- Input jumlah barang
- Input lanjut / cancel
- Pemilihan metode pembayaran
- Memasukkan kode pembayaran jika nontunai dan uang jika tunai



PROSES

- Fungsi stok barang
- Fungsi refill
- Fungsi hitung kembalian
- Menghitung harga total
- Percabangan pilihan barang
- Percabangan lanjut/cancel
- Percabangan metode pembayaran



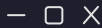
OUTPUT

- Fungsi QR code
- Barang yang dibeli
- Kembalian jika ada
- Ucapan terimakasih





KAMUS



Kita awali kode dengan mendeklarasikan variable-variable yang akan digunakan dalam program ini.

```
produk = arr of string
```

```
stok_barang, pemesanan, list_of_pecahan = arr  
of int
```

```
kode, harga_total, uang, kembalian = int
```

```
idx, i = int
```

```
pecahan = dictionary of (string: int)
```

```
cancel_atau_lanjut, metode = string
```

```
kode_rahasia = string
```



Inisialisasi produk dan sistem stok barang

Kode tentu saja diawali dengan inisialisasi. Inisialisasi terdiri atas inisialisasi array barang dan pembuatan file stok barang. File stok barang berfungsi untuk menyimpan data yang berubah setelah vending machine digunakan misalkan setelah barang diambil 1, barang akan benar-benar berkurang 1.

```
19 # Inisialisasi Array Nama Produk dan Harga Produk
20 produk = ["Chitatho", "Yakul", "Akua", "The pucuk", "Nescafo", "Poup mie", "Cola", "No Green Tea",
21 |         "Hypermilk", "Pocahri", "Pocki", "Silverking"]
22
23 harga = [2000, 3000, 4000, 5000, 6000, 6000, 7000, 7000, 7000, 8000, 11000, 14000]
24
25 pemesanan = [0 for i in range(len(produk))]
26
27 # Membuat File Text yang Berisi Daftar Stok Barang Jika Belum Ada
28 if not os.path.exists("vend_mach_data.txt"):
29     file = open("vend_mach_data.txt", "x")
30     file = open("vend_mach_data.txt", "a")
31     for i in range(12):
32         file.write("10\n")
33
34 # Read Text File Stok Barang
35 file = open("vend_mach_data.txt", "r")
36 stok_barang = file.readlines()
37 file.close()
38
39 # Membenarkan Element Stok Barang Menjadi Integer
40 for i in range(len(stok_barang)):
41     stok_barang[i] = int(stok_barang[i].strip())
42
```

Fungsi / subprogram

Dibutuhkan fungsi / subprogram untuk menyingkat suatu kode yang akan dipakai berkali-kali. Berikut fungsi antarmuka dan fungsi QR code.

```
45 # Formatting
46 def justify(separator, number, word1, word2, word3):
47
48     separator = str(separator)
49     number = str(number)
50     word1 = str(word1)
51     word2 = str(word2)
52     word3 = str(word3)
53
54     print(number, word1, end = "")
55     for i in range(40 - len(number) - len(word1) - len(word2)):
56         print(separator, end = "")
57     print(word2, word3)
58
59 # Tampilkan qr code
60 def draw_qr():
61     print(" ")
62     print(" ")
63     print(" ")
64     print(" ")
65     print(" ")
66     print(" ")
67     print(" ")
68     print(" ")
69     print(" ")
70     print(" ")
71     print(" ")
72     print(" ")
73     print(" ")
74     print(" ")
75     print(" ")
76     print()
```



Fungsi / subprogram

Dibutuhkan fungsi / subprogram untuk menyingkat suatu kode yang akan dipakai berkali-kali. Berikut fungsi antarmuka dan fungsi refill barang. Saat user memasukan kode rahasia, apabila kode rahasia benar, fungsi refill barang akan dipanggil.

```

109
110 # Update Stok Barang
111 file = open("vend_mach_data.txt", "w")
112 file.write("")
113 file.close()
114
115 file = open("vend_mach_data.txt", "a")
116 for i in range(12):
117     file.write(f"{stok_barang[i]}\n")
118 file.close()
119
120 # Selesai
121 draw_vending_machine()
122 print("Telah keluar mode refill, stok barang sudah ter-update.")
123

```

```

78 # Tampilan Vending Machine
79 def draw_vending_machine():
80     justify(" ", "No.", "Produk", "Harga", "Stok")
81     for idx in range(12):
82         justify("-", idx+1, produk[idx], harga[idx], stok_barang[idx])
83     print()
84
85 # Mode Refill
86 def mode_refill():
87     draw_vending_machine()
88     print("Selamat datang di mode refill!")
89
90     while True:
91         kode = int(input("Pilih kode produk mana yang ingin di refill (0 untuk keluar, 13 untuk refill semuanya): ")) - 1
92         if kode == -1:
93             break
94
95         elif kode == 12:
96             for i in range(12):
97                 stok_barang[i] + 10
98             print("Berhasil! Semua produk sudah di refill.")
99             print()
100             break
101
102         elif 0 < kode < 11:
103             stok_barang[kode] + 10
104             print(f"Berhasil! {produk[kode]} sudah di refill.")
105             print()
106
107         else:
108             kode = int(input("Error, masukkan kode 0-13: "))
109

```



Fungsi / subprogram

Dibutuhkan fungsi / subprogram untuk menyingkat suatu kode yang akan dipakai berkali-kali. Berikut fungsi hitung kembalian yang berfungsi untuk menentukan pecahan-pecahan uang untuk dikembalikan kepada user ketika mereka membayar lebih atas produk yang dibeli.

```
124 # Hitung Kembalian
125 def hitung_kembalian(kembalian):
126
127     # Dictionary kembalian
128     pecahan = {"pecahan_50": 0, "pecahan_20": 0, "pecahan_10": 0, "pecahan_5": 0,
129               | "pecahan_2": 0, "pecahan_1": 0
130     }
131
132     # Hitung masing-masing pecahan
133     pecahan["pecahan_50"] = kembalian // 50_000
134     kembalian %= 50_000
135
136     pecahan["pecahan_20"] = kembalian // 20_000
137     kembalian %= 20_000
138
139     pecahan["pecahan_10"] = kembalian // 10_000
140     kembalian %= 10_000
141
142     pecahan["pecahan_5"] = kembalian // 5_000
143     kembalian %= 5_000
144
145     pecahan["pecahan_2"] = kembalian // 2_000
146     kembalian %= 2_000
147
148     pecahan["pecahan_1"] = kembalian // 1_000
149
150     return list(pecahan.values())
151
```



Algoritma Utama



Memanggil fungsi antarmuka untuk menampilkan menu vending machine, melakukan print perkenalan, serta inisialisasi harga total awal dan kode rahasia.

```
153
154 # Print UI
155 os.system("CLS")
156 draw_vending_machine()
157 print("Selamat datang di Vending Machine Kelompok 5!")
158 print("Mau beli apa?")
159 print("Masukkan 0 untuk selesai dan lanjut ke pembayaran")
160
161 # Inisialisasi harga awal
162 harga_total = 0
163
164 # Inisialisasi kode rahasia
165 kode_rahasia = 123456789
166 # Sebaiknya kodenya [ ] tetapi harus dikurangi 1 karena input kode dikurangi 1
167
```



Algoritma Utama

Membuat loop input untuk user memilih apa saja yang mau dibeli oleh user dan apabila user adalah admin yang ingin melakukan refill stok barang bisa memasukkan kode rahasia ketika ditanya kode stok barang.

```
168 # Input terus
169 while True:
170
171     # Input kode
172     kode = input("Masukkan kode barang yang ingin dibeli (0 untuk selesai dan lanjut ke pembayaran): ")
173
174     # Jika Kode rahasia untuk mode refill
175     if kode == kode_rahasia:
176         os.system("cls")
177         mode_refill()
178         exit()
179
180     # Jika kode salah
181     if kode not in ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"]:
182         print()
183         print(f"Error, kode tidak terbaca!")
184         print()
185
186     else:
187         # kode nya dijadiin integer
188         kode = int(kode) - 1
189
190     # Jika customer memasukkan 0 tetapi belum membeli apa-apa
191     if kode == -1 and harga_total == 0:
192         print("Anda belum beli apa-apa... ")
193         kode = int(input("Apakah tetap ingin quit? Masukkan 0 kembali untuk keluar: "))
194         if kode == 0:
195             print("Terima kasih telah menggunakan vending machine ini!")
196             exit()
197         else:
198             pass
199
```



Algoritma Utama



Percabangan dari loop input user apabila user sudah selesai memilih barang dan jika barang yang dipilih user stoknya habis.

```
200     # Customer sudah selesai, lanjut bayar
201     elif kode == -1:
202         print()
203         break
204
205     # Pemesanan benar
206     elif 0 <= kode <= 11:
207
208         # Jika stoknya gaada
209         if stok_barang[kode] == 0:
210             print(f"Mohon maaf, stok {produk[kode]} sedang habis, silakan beli produk lain.")
211
```





Algoritma Utama



Input jumlah barang yang ingin dibeli user, algoritma harga total, dan percabangan jika input user invalid.

```
212 # Input jumlah yang ingin dibeli hingga benar
213 else:
214     jumlah = int(input(f"Masukkan jumlah dari {produk[kode]} yang ingin dibeli: "))
215
216     # Jika stok memenuhi
217     if stok_barang[kode] >= jumlah:
218
219         # Tambah ke harga total
220         harga_total += jumlah * harga[kode]
221
222         # Kurangi stok dengan jumlah yang dibeli
223         stok_barang[kode] -= jumlah
224
225         # Tambahkan ke array pemesanan
226         pemesanan[kode] += jumlah
227
228         # Print harga total
229         print(f"Harga total = {harga_total}")
230
231     # jika stok tidak cukup
232     else:
233         print(f"Mohon maaf, stok {produk[kode]} tidak mencukupi, silakan beli produk lain.")
234
235 # Kode tidak sesuai
236 else:
237     print("Error, kode yang Anda masukkan tidak sesuai. Silakan memasukkan input kembali.")
238     sleep(0.5)
239
240 print()
241
```



Algoritma Utama

Metode pembayaran tunai. User akan diminta untuk membayar apa yang sudah dibeli dengan uang kelipatan Rp. 10.000,- dan jika tidak cukup atau invalid akan diulang sampai user memasukkannya dengan benar

```
266
267 # Pilih metode pembayaran
268 metode = input("Masukkan metode pembayaran tunai/nontunai: ")
269 while metode not in ["tunai", "nontunai"]:
270     metode = input("Error, masukkan metode pembayaran tunai/nontunai: ")
271 print()
272
273 if metode == "tunai":
274
275     # Input uang
276     uang = int(input("Masukkan nominal uang pembayaran (hanya menerima kelipatan Rp10.000): "))
277
278     # Jika uang masih salah:
279     while (uang % 10_000 != 0) or (uang <= harga_total):
280
281         # Jika uang bukan kelipatan 10.000
282         if uang % 10_000 != 0:
283             uang = int(input("Error, masukkan nominal uang dengan kelipatan Rp10.000: "))
284
285         # Jika uang tidak mencukupi
286         else:
287             uang = int(input("Waduh, uang Anda tidak cukup, silakan memasukkan kembali nominal uang: "))
288
```



Algoritma Utama



Menghitung kembalian jika user memasukkan uang yang lebih dari harga produk dengan memanggil fungsi hitung kembalian.

```
288
289     # Hitung kembalian
290     kembalian = uang - harga_total
291     pecahan = ["Rp50.000", "Rp20.000", "Rp10.000", "Rp5.000", "Rp2.000", "Rp1.000"]
292
293     if kembalian != 0:
294         list_pecahan = hitung_kembalian(kembalian)
295
296     # Ini biar terlihat real, terdapat loading screen
297     print("Pembayaran sedang diproses.", end= " ")
298     for i in range(3):
299         print(".", end= " ")
300         sleep(0.7)
301     print()
302     print()
303
304     # Output pecahan uang
305     if kembalian != 0:
306         print(f"Kembalian Anda adalah Rp{kembalian} dengan rincian kembalian berikut: ")
307         for i in range(len(pecahan)):
308             if list_pecahan[i] != 0:
309                 justify("", str(list_pecahan[i]) + "x", pecahan[i], "", "")
310         print()
311     else:
312         print("Berhasil, Anda membayar dengan uang pas!")
313
```





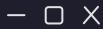
Algoritma Utama



Jika nontunai, memanggil fungsi QR code.

```
313  
314 # Memanggil fungsi QR code  
315 elif metode == "nontunai":  
316     draw_qr()  
317     print(f"Silakan menscan QR code diatas.")  
318     input(f"Masukkan kode pembayaran: ")  
319  
320     print("Pembayaran sedang diproses.", end= " ")  
321     for i in range(3):  
322         print(".", end = " ")  
323         sleep(0.7)  
324     print()  
325
```





Algoritma Utama



Output terimakasih dan algoritma update stok barang untuk memperbaharui data bahwa ada stok barang yang sudah berkurang.

```
326 # Output
327 print("Selesai! Terima kasih telah menggunakan vending machine ini! Selamat menikmati!")
328 print("Selamat datang kembali")
329
330 # Update Stok Barang
331 file = open("vend_mach_data.txt", "w")
332 file.write("")
333 file.close()
334
335 file = open("vend_mach_data.txt", "a")
336 for i in range(12):
337     file.write(f"{stok_barang[i]}\n")
338 file.close()
```





<Terima Kasih!>





Referensi



- slidesgo.com

