D. Instalasi Library

Setelah melakukan instalasi Python, VSCode dan Jupyter Notebook, yang perlu kamu install selanjutnya adalah Library. Library diibaratkan sebuah add-on untuk Python. Cara penginstallannya adalah sebagai berikut :

1. Buka terminal pada VSCode atau CMD
2. Ketik *"python -m pip install ..."* tanpa tanda kutip
3. Titik-titik diisi dengan library yang ingin diinstall. Misalnya untuk menginstall pandas, maka "python -m pip install pandas"

Beberapa library yang perlu kamu install adalah sebagai berikut :

1. Pandas

   Library ini menyediakan struktur data dan alat analisis data yang mudah digunakan dan efisien. Ini sangat berguna untuk membaca, mengolah, dan menganalisis data dari berbagai sumber. Library ini akan sangat berguna pada materi analisis data. Cara installnya adalah dengan mengetikkan pada CMD : *"python -m pip install pandas"*
2. Matplotlib

   Library ini memungkinkan pengguna untuk membuat visualisasi data yang menarik dan informatif, termasuk grafik garis, grafik batang, dan banyak lagi. Library ini sangat penting untuk sub-bab visualisasi data. Cara installnya adalah dengan mengetikkan pada CMD: *"python -m pip install matplotlib"*
3. Numpy

   Library ini menyediakan objek array yang efisien untuk komputasi numerik dan matematika. Pada buku ini tidak akan diajarkan mengenai numpy, karena pada perkuliahan tidak diajarkan. Kamu bisa mengeksplor sendiri melalui internet maupun ChatGPT terkait penggunaan numpy. Cara installnya adalah : *"python -m pip install numpy"*

E. Cara Penulisan Kode Dengan Baik pada Pengenalan Komputasi

Penulisan kode program sebetulnya tidak ada ketentuan khusus kecuali sintaks dari program itu sendiri. Pada buku ini, setiap pembahasan tidak akan pula dituliskan format-format tertentu. Namun saat aku mengambil mata kuliah ini, aku diajarkan beberapa cara penulisan khusus untuk mata kuliah pengenalan

komputasi ini. Oleh karena itu, akan aku ajarkan bagaimana cara menulis kode dengan baik untuk mata kuliah Pengenalan Komputasi.

1. **Identitas**

   Pertama, kamu harus menulis identitas seperti nama, nim, dan sebagainya. Penulisan ini harus diawali dengan tanda pagar #. Tanda ini digunakan untuk menulis komentar dalam Python. Untuk tugas-tugas biasa, format yang biasa digunakan adalah sebagai berikut :

```
1. # NIM/Nama   : 16421359/Fachri Misykatul Anwar
2. # Tanggal    : 17 Oktober 2021
3. # Minggu     : 8
4. # Tugas      : Latihan 1
```

   Tulis NIM/NAMA pada baris pertama. Baris kedua berisi tanggal pengerjaan tugas, baris ketiga berisi minggu perkuliahan, dan baris keempat adalah judul tugas.

   Sedangkan untuk praktikum, format yang biasa digunakan adalah sebagai berikut :

```
1. # NIM/Nama   : 16421359/Fachri Misykatul Anwar
2. # Tanggal    : 7 Oktober 2021
3. # Kode Shift: 4.5
4. # Deskripsi : (Problem 01) Menghitung luas segitiga
```

   Tulis NIM/NAMA pada baris pertama. Baris kedua berisi tanggal pengerjaan tugas, baris ketiga diisi kode shift praktikum, dan baris keempat diisi dengan deskripsi soal.

2. **Ide/Rencana**

   Hal kedua yang perlu kamu tulis adalah ide/rencana. Ide yang dimaksud adalah bagaimana cara berpikir kamu dalam menyelesaikan soal tersebut. Hal ini ditulis saat pengerjaan praktikum. Namun untuk tugas-tugas perkuliahan biasanya tidak ditulis. Tetapi akan lebih baik apabila kamu menuliskannya juga. Contoh penulisannya adalah sebagai berikut :

```
1. # Ide / Rencana
2. """
3. Untuk menghitung luas segitiga, pertama kita harus menginput alas (a)
dan tinggi (t) sebagai bilangan real (float). Setelah itu buat variabel
L dan hitung luas segitiga dengan rumus L = (a*t)/2. Barulah kita
keluarkan output dari luas segitiga tersebut.
4. """
```


Dipindai dengan CamScanner

Penulisan diawali dengan judul Rencana / Ide. Setelah itu buat 3 tanda petik pada baris kedua dan keempat. Pada baris ketiga, kamu bisa menuliskan dan menceritakan rencana pengerjaan kamu.

3. **Kamus**

Hal ketiga yang perlu kamu tulis adalah kamus. Kamus ini digunakan untuk menjelaskan tipe data sebuah variabel yang terdapat pada program kamu. Contoh penulisannya adalah sebagai berikut :

```
1. # KAMUS
2. # N,M :int
3. # A : Matrix NxM of int
```

4. **Algoritma**

Setelah semuanya sudah tertulis dengan rapih, maka kamu sudah bisa menuliskan program dengan diawali dengan # Algoritma. Berikut ini adalah contoh penulisan program praktikum menghitung luas segitiga :

```
1. # NIM/Nama  : 16421359/Fachri Misykatul Anwar
2. # Tanggal   : 1 Januari 2023
3. # Kode Shift: 4.5
4. # Deskripsi : (Problem 01) Menghitung luas segitiga
5.
6. # IDE / RENCANA
7. """
8. Untuk menghitung luas segitiga, pertama kita harus menginput alas
   (a) dan tinggi (t) sebagai bilangan real (float). Setelah itu buat
   variabel L dan hitung luas segitiga dengan rumus L = (a*t)/2.
   Barulah kita keluarkan output dari luas segitiga tersebut.
9. """
10.
11. # KAMUS
12. # a,t : float
13.
14. # Algoritma
15. a = float(input("Masukkan alas : "))
16. t = float(input("Masukkan tinggi: "))
17. L = (a*t)/2
18. print(f"Luas segitiga : {L}")
```

F. Keterangan Pada Buku

Soal-soal latihan pada buku ini beberapa diambil dari soal praktikum, UTS, dan UAS tahun-tahun sebelumnya. Terutama soal-soal tahun 2021, tahun aku TPB. Beberapa soal yang diambil juga diubah narasinya, namun tidak mengubah tujuan soal. Selebihnya, aku sendiri yang menulisnya. Pada soal UTS dan UAS, terdapat

Dipindai dengan CamScanner

BAB I : Computing System, Network, dan Internet

KOMPUTER DAN GREEN COMPUTING

Komputer adalah alat yang dipakai untuk mengolah data menurut prosedur yang telah dirumuskan. Green computing adalah usaha menggunakan sumber daya komputasi secara efisien. Misalnya meminimalkan konsumsi listrik, meminimalkan penggunaan kertas, memperpanjang masa pakai perangkat keras.

HARDWARE

Hardware / perangkat keras adalah komponen fisik kompupter. Hardware komputer terbagi menjadi 4, input device, output device, CPU, serta memory dan storage device. Contoh input device adalah keyboard dan mouse. Contoh output device adalah monitor dan speaker. CPU (Central Processing Unit) melakukan manipulasi aritmatika dalam logic terhadap data. Memory dan Storage Device terbagi 4, yaitu RAM (Random Access Memory), ROM (Read-only Memory), CMOS (Complementary metal-oxide semiconductor), dan Flash memory.

SOFTWARE

Software/Perangkat Lunak adalah data yang diprogram dan diformat secara digital untuk tujuan tertentu. Software terdiri dari sistem operasi dan perangkat lunak utilitas. Contoh software utilitas berbasis web adalah Google Apps, Email, Wikipedia, amazon, ebay, facebook, twitter, instagram. Contoh software utilitas general adalah office word, powerpoint, excel, chrome. Device driver adalah program yang memungkinkan hardware berkomunikasi dengan komputer. Sistem operasi adalah perangkat lunak sistem yang mengelola perangkat keras dan sumber daya komputer lain yang memberikan layanan pada program-program komputer. Contohnya adalah Windows, MacOS, Android, iOS.

JARINGAN

Jaringan adalah suatu yang menghubungkan komputer satu sama lain. Hardware jaringan antara lain routers, bridges, repeater, Network Interface Card / Ethernet Card, Hub. Jenis-jenis jaringan antara lain Local Area Network (LAN), Metropolitan Area Networks (MAN), Wide Area Networks (WAN), Wireless Networks, Home Networks, dan Internet.

INTERNET

Internet (interconnected network) adalah jaringan komputer yang saling terhubung secara global.Sejarah internet dimulai dari ARPANET pada 1969 dan bekerja secara peer-to-peer, dibubarkan pada 1990. TCP/IP (Transmission Control Protocol / Internet Protocol) adalah bahasa dari internet yang memungkinkan komunikasi antar jaringan. TCP memecah suatu pesan menjadi paket-paket, dan IP adalah alamat untuk paket diantarkan. DNS (Domain Name System) adalah menerjemahkan alamat IP menjadi suatu nama. ISP (Internet Service Providers) adalah penyedia koneksi internet. Cloud Computing adalah ketersediaan sumber daya sistem komputer sesuai permintaan, terutama penyimpanan data dan daya komputasi, tanpa pengelolaan aktif langsung oleh pengguna.

Dipindai dengan CamScanner

SOAL-SOAL

1. Sebutkan contoh-contoh computing devices
2. **(UTS 2021)** Jelaskan apa yang disebut dengan *green computing*
3. **(UTS 2021)** Jelaskan 2 (dua) contoh aplikasi dari *green computing*
4. Jelaskan operasi dasar yang dilakukan oleh komputer
5. Jelaskan apa itu hardware dan macam-macamnya
6. Jelaskan apa yang dimaksud dengan software/perangkat lunak
7. Jelaskan apa itu algoritma dan program
8. **(UTS 2021)** Sebutkan 2 (dua) contoh *software*/perangkat lunak, baik yang tersedia *online* (berbentuk aplikasi web atau *mobile*) maupun yang *offline* (aplikasi desktop) yang mendukung pelaksanaan kuliah *online*. Untuk masing-masing contoh, jelaskan bagaimana perangkat lunak tersebut mendukung Anda dalam melaksanakan kuliah *online*
9. Perkembangan aplikasi berbasis website semakin pesat dan mulai menggantikan software desktop. Misalnya Microsoft Word, Power Point, dan Excel kini sudah tersedia versi web. Apakah kehadiran aplikasi ini akan menggantikan software desktop sepenuhnya?
10. Jelaskan apa itu sistem operasi dan sebutkan (minimal 3) contoh sistem operasi
11. Jelaskan apa itu jaringan komputer dan jenis-jenisnya
12. **(UTS 2021)** Sebutkan dan jelaskan 2 (dua) contoh ancaman terhadap privasi karena perkembangan teknologi *computing* dan untuk masing-masing contoh, berikan 1 (satu) contoh tindakan yang dapat dilakukan oleh pribadi untuk menghindari/mengatasinya.
13. **(UTS 2018)** Suatu hari saat akan praktikum Pengenalan Komputasi, Anda menemukan komputer yang akan digunakan masih hidup dan pada *browser* masih terlihat aplikasi media sosial terbuka. Anda mengenal pemilik akun yang sedang aktif pada aplikasi tersebut dan dia kebetulan adalah orang yang sangat tidak Anda sukai. Orang tersebut sering menggunakan akun media sosial ini untuk melakukan perundungan (*cyber-bullying*) kepada orang lain, termasuk Anda. Anda berpikir jika Anda melakukan sesuatu pada akun tersebut, misalnya mengirimkan pesan yang tidak pantas, menghapus *file* dan gambar, dll, banyak orang yang akan merasa senang, bahkan berterima kasih pada Anda. Lagipula salah dia sendiri tidak log-out dengan baik dari aplikasi tersebut di komputer umum. Apa yang akan Anda lakukan pada situasi ini?

Dipindai dengan CamScanner

PEMBAHASAN

1. Beberapa contoh computing devices adalah :
   - Personal Computer (PC), yaitu komputer pribadi seperti desktop atau laptop
   - Konsol game, misalnya PlayStation (PS) atau XBox
   - Smartphone dan Tablet
   - Kalkulator
   - Server

2. Green computing adalah konsep yang dirancang untuk mempromosikan penggunaan teknologi informasi dan komunikasi (TIK) yang lebih ramah lingkungan dan bertanggung jawab secara sosial. Tujuannya adalah untuk mengurangi dampak negatif TIK pada lingkungan dan mendorong praktik yang lebih berkelanjutan.

3. Contoh dari Green Computing adalah sebagai berikut :
   - Virtualisasi : Berkembangnya teknologi layar sentuh memungkinkan pengetikan yang tidak menggunakan lagi keyboard fisik. Pengurangan penggunaan keyboard fisik dapat mengurangi sampah elektronik serta menghemat energi.
   - Cloud Computing : Cloud computing memungkinkan organisasi untuk menyimpan dan mengakses data dan aplikasi mereka di jaringan yang disebut "cloud" atau awan. Cloud computing dapat mengurangi penggunaan perangkat keras yang mahal dan meminimalkan penggunaan listrik yang berlebihan.

4. Operasi dasar yang dilakukan oleh komputer ada 4 :
   - Menerima input : Yaitu menerima informasi dari luar komputer
   - Memproses informasi : Yaitu menerapkan algoritma terhadap informasi yang diperoleh
   - Mengeluarkan output : Yaitu mengomunikasikan informasi keluar komputer
   - Menyimpan informasi : Yaitu menyimpan dan mengambil informasi dari memori/penyimpanan komputer

5. Perangkat keras (hardware) adalah sekumpulan alat fisik agar komputer dapat berjalan dengan baik. Ada 4 macam hardware :
   - Input device : yaitu perangkat yang menerima input dari pengguna. Misalnya keyboard, mouse, dll
   - Output device : yaitu perangkat yang menuliskan hasil keluaran program. Misalnya printer, monitor, speaker, dll
   - Microprocessor (CPU) : yaitu bagian yang mengendalikan dan melakukan pemrosesan dari input ke output.
   - Memory dan Storage Devices : yaitu piranti penyimpanan untuk menyimpan data dalam komputer. Misalnya adalah RAM, ROM, CMOS, dll.

6. Perangkat lunak (software) adalah kumpulan dari program, data dan dokumentasi yang memungkinkan pengguna mengomunikasikan suatu persoalan kepada komputer dan komputer memberikan solusinya pada pengguna.

7. Algoritma adalah sekumpulan instruksi logis atau langkah-langkah penyelesaian dari suatu persoalan. Program adalah implementasi dari algoritma dalam bentuk kode pemrograman yang dapat dijalankan oleh komputer.

Dipindai dengan CamScanner

8. Ada beberapa software / perangkat lunak yang dapat mendukung perkuliahan *online* :
   - Google Classroom : Google classroom adalah aplikasi berbasis web yang memungkinkan guru untuk memposting materi dan tugas, dan memungkinkan siswanya untuk mengumpulkan tugas dan memperoleh materi dari guru.
   - Zoom : Zoom adalah aplikasi desktop yang digunakan untuk konferensi video. Zoom digunakan untuk kegiatan pembelajaran daring dan jarak jauh. Zoom memiliki fitur *sharescreen*, breakout rooms, dan polling yang berguna untuk kegiatan pembelajaran.

9. Aplikasi berbasis website dan desktop memiliki kekurangan dan kelebihannya masing-masing. Aplikasi berbasis website memiliki keunggulan aksesibilitasnya yang tidak terbatas pada sistem operasi tertentu, memudahkan kolaborasi dan kerja sama tim, dan perawatan yang lebih mudah karena dikelola oleh penyedia layanan. Namun, ada kelemahan yang dimiliki aplikasi berbasis website. Aplikasi website masih memiliki banyak keterbatasan fungsi dibandingkan aplikasi desktop yang lebih canggih. Aplikasi website juga harus membutuhkan internet yang mana apabila tidak ada internet, aplikasi tidak dapat berjalan. Oleh karena itu, aplikasi berbasis website tidak akan sepenuhnya menggantikan software desktop.

10. Sistem operasi adalah perangkat lunak sistem yang mengelola perangkat keras dan sumber daya komputer yang lain dan memberikan layanan pada program-program komputer. Contoh-contoh dari sistem operasi adalah Microsoft Windows, Apple MacOS, dan Linux.

11. Jaringan komputer adalah koneksi antara dua atau lebih perangkat komputer agar dapat berkomunikasi dan berbagi sumber daya seperti file, printer, dan koneksi internet. Jenis-jenis jaringan komputer antara lain:
    - Local Area Network (LAN) : Jaringan yang mencakup area yang terbatas seperti satu gedung atau kantor yang terhubung melalui kabel atau wireless.
    - Metropolitan Area Network (MAN) : Jaringan yang mencakup area yang lebih luas dari LAN, seperti kota atau wilayah geografis yang lebih besar.
    - Wide Area Network (WAN) : Jaringan yang mencakup area yang sangat luas seperti negara atau benua.
    - Internet : Jaringan komputer global yang terhubung melalui infrastruktur kabel serat optik, satelit, dan teknologi nirkabel.

12. Berikut dua contoh ancaman terhadap privasi :
    - Penyalahgunaan data pribadi : Perkembangan teknologi computing, seperti Big Data dan Internet of Things (IoT), telah menyebabkan pengumpulan data yang besar dan kompleks tentang pengguna. Hal ini dapat meningkatkan risiko penyalahgunaan data pribadi, seperti pencurian identitas, penipuan finansial, atau penggunaan data pribadi untuk tujuan yang tidak sah. Tindakan yang perlu dilakukan adalah dengan memperhatikan pengaturan privasi di platform online dan aplikasi dalam mengumpulkan data pribadi. Kita tidak boleh asal setuju dan terima setiap dialog yang ditampilkan mengenai privasi kita.

Dipindai dengan CamScanner

- Serangan siber : Serangan siber oleh hacker dapat mengancam privasi individu dengan mencuri data pribadi kita seperti email, kata sandi dan bahkan informasi kartu kredit. Misalnya saja penyebaran apk dengan modus lacak resi paket, sedangkan aplikasi tersebut disisipi program pengintai yang mencuri data pribadi kita. Yang harus kita lakukan adalah berhati-hati dalam menggunakan software. Gunakan aplikasi yang terpercaya. Jangan asal mengunduh dan menginstall aplikasi. Jangan gunakan aplikasi bajakan, gunakanlah aplikasi resmi yang tersedia di playstore maupun appstore (untuk mobile).

13. Sebagai seorang individu yang bertanggung jawab dan menghormati etika penggunaan teknologi, kita tidak boleh melakukan tindakan yang merugikan orang lain, dan segera menutup aplikasi media sosial pada komputer tersebut. Jika memang pemilik akun tersebut kerap melakukan cyber bullying, sebaiknya kita laporkan kepada pihak yang berwenang, tidak boleh main hakim sendiri.

Dipindai dengan CamScanner

BAB II : Computational and Algorithmic Thinking

Berpikir komputasional adalah proses pemikiran yang terlibat dalam merumuskan masalah dan mengungkapkan solusinya sedemikian rupa sehingga komputer – manusia atau mesin – dapat melakukannya secara efektif. **Empat pilar berpikir komputasional** adalah :

- **Dekomposisi** adalah menyelesaikan sebuah masalah dengan cara membagi masalah menjadi sub-masalah yang lebih kecil, kemudian memecahkan masalah-masalah yang lebih kecil tersebut secara terpisah.
- **Pengenalan pola** adalah mengenali kesamaan/perbedaan karakteristik dari berbagai permasalahan/fenomena yang berbeda tetapi memiliki keterkaitan.
- **Abstraksi dan generalisasi** adalah menghilangkan bagian-bagian dari sebuah permasalahan yang tidak penting (abstraksi) untuk mendapatkan solusi umum dalam memecahkan masalah yang sejenis (generalisasi).

Algoritma adalah himpunan prosedur langkah per langkah untuk menyelesaikan suatu [sub]persoalan. Algoritma dapat ditulis dengan pseudocode dan flowchart. Pemrograman (*programming*) adalah salah satu bentuk penyelesaian (*problem solving*) dimana persoalan serta solusinya direpresentasikan dalam bentuk yang bisa diproses oleh komputer. Secara umum, terdiri atas langkah-langkah sebagai berikut :

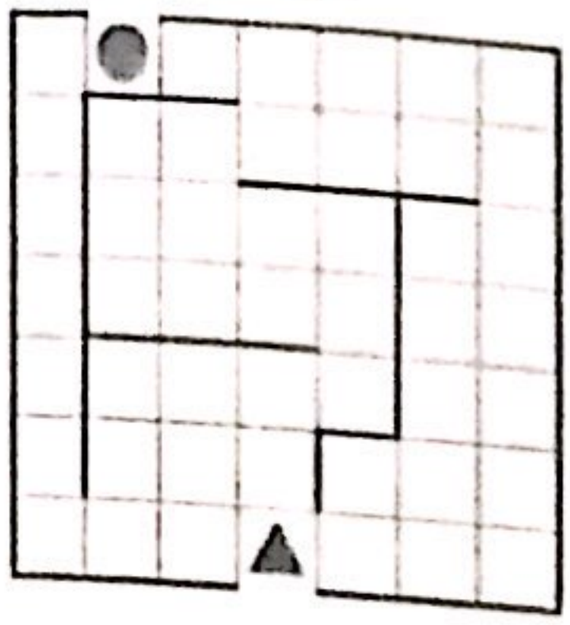
- Memahami persoalan
- Menyusun rencana untuk menyelesaikan persoalan
- Menyusun solusi berdasarkan rencana
- Mengevaluasi solusi

Programmer mengubah algoritma menjadi kode program komputer dengan menggunakan bahasa pemrograman. Proses untuk menuliskan kode program berdasarkan algoritma disebut sebagai coding. File hasil menuliskan kode program disebut source code (kode sumber). Setiap pernyataan dalam algoritma ditranslasikan secara detil ke dalam kode program. Compiler/interpreter akan mentranslasi kode program dalam bahasa pemrograman tertentu menjadi bentuk yang dipahami oleh komputer.

Setiap komputer memproses instruksi dalam bahasa mesin (machine language). Programmer menggunakan bahasa pemrograman tingkat tinggi (high-level languages) untuk menuliskan kode program. Misalnya adalah Pascal, C/C++, Python, Matlab, dan sebagainya.

Dipindai dengan CamScanner

1. **(UTS 2018)** Jelaskan mengapa kemampuan berpikir komputasional menjadi salah satu skill yang penting di abad 21!
2. **(UTS 2018)** Untuk menyelesaikan tugas kuliah, dibutuhkan perangkat lunak yang berbayar yang cukup mahal. Kampus menyediakan perangkat lunak tersebut di laboratorium komputer, tetapi Anda harus berbagi dengan mahasiswa lain dan tidak dapat menggunakannya setiap saat. Anda tahu bahwa Anda dapat membeli versi bajakan dari perangkat lunak tersebut dengan cukup murah dan juga mudah didapatkan di internet. Apa yang akan Anda lakukan? *crack*
3. **(Contoh 1)** Mira perlu menemukan jalan untuk keluar dari sebuah labirin dan meminta anda untuk memberikan arahan. Dia memasuki labirin dari bawah (segitiga hitam) dan harus mencapai pintu keluar pada bagian atas (lingkaran merah besar).



Namun, Mira hanya dapat mengingat empat gerakan berikut :

Kode Gerakan	Artinya	Ilustrasi
A	Berjalan satu langkah maju dan menghadap ke kiri	
B	Berjalan satu langkah maju dan menghadap ke kanan	
C	Berjalan dua langkah maju dan menghadap ke kiri	
D	Berjalan dua langkah maju dan menghadap ke kanan	

Mira bisa membentuk sebuah gerakan panjang yang dibentuk dari empat gerakan A, B, C, atau D. Satu gerakan bisa diulang beberapa kali. Urutan gerakan ini dapat membawa Mira keluar dari labirin dengan dua kali perulangan. Urutan gerakan manakah yang jika diulang dua kali oleh Mira dapat membuatnya mencapai pintu keluar?

4. **(Contoh 2)** Ada tujuh (7) siswa yang gemar membaca buku dan mereka membentuk klub untuk berbagi buku. Jika ada satu buku baru diperoleh (dan dibaca) seorang siswa, kemudian ia akan meneruskan meminjamkan ke anggota klub lainnya dengan cara berikut :

Dipindai dengan CamScanner

- Tidak setiap siswa menjadi sahabat siswa lainnya, maka seorang siswa hanya meneruskan meminjamkan buku ke siswa yang bersahabat dengannya.
- Jika seorang siswa mempunyai beberapa sahabat, maka sahabat yang paling muda yang akan dipinjami terlebih dulu, yang belum pernah meminjam buku itu.
- Kalau semua sahabatnya sudah pernah meminjamnya, maka ia akan mengembalikan ke siswa yang sebelumnya meminjamkan buku itu kepadanya.

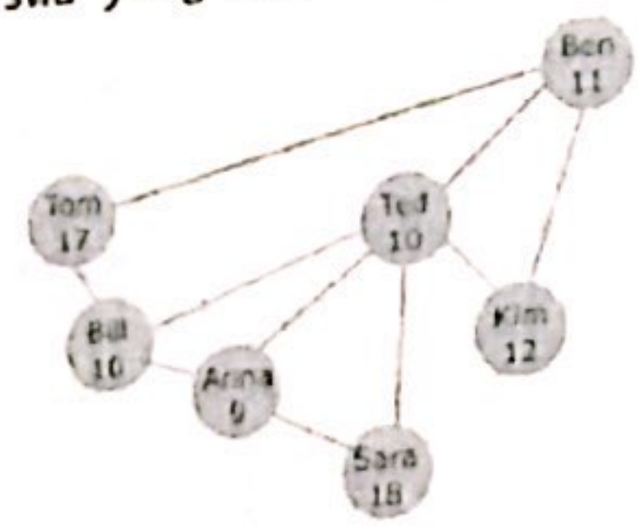
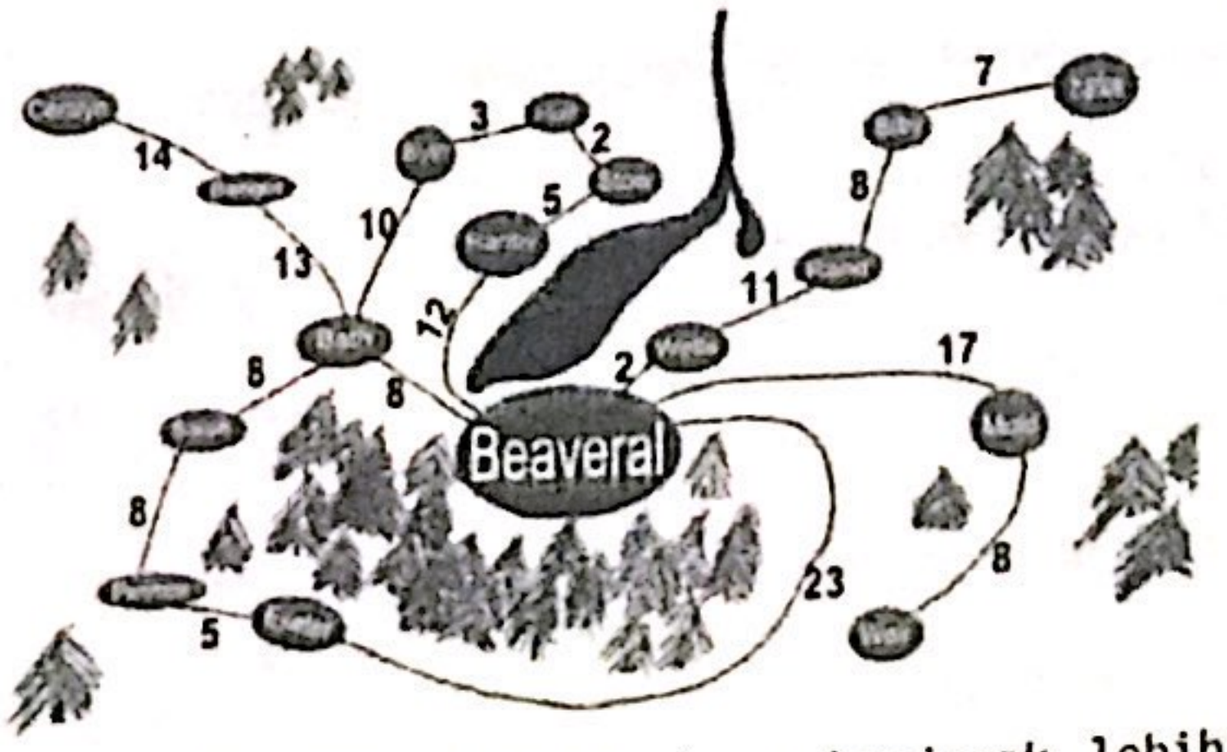


Diagram di atas menunjukkan tujuh siswa dan garis-garis menunjukkan hubungan "sahabat" itu. Setiap simpul berisi informasi nama dan umur. Ben selesai membaca sebuah buku baru dan ingin berbagi dengan semua anggota klub dan selain Ben belum ada yang pernah membacanya. Siapa yang akan menjadi pembaca terakhir dari buku tersebut?

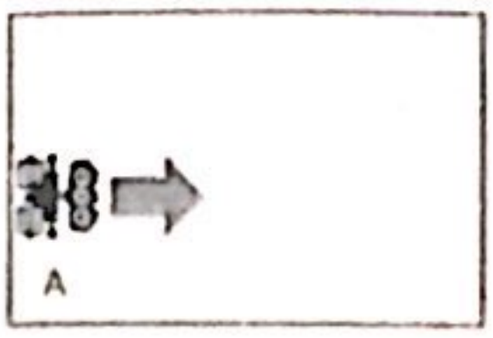
5. (Contoh 6) Patra tinggal di desa Beaveral. Ia ingin mengundang teman-temannya untuk merayakan ulang-tahunnya. Namun ia hanya mengundang teman-teman yang tinggal tidak lebih dari 20 km dari rumahnya agar tak kemalaman pulang. Jarak rumah teman-temannya dimunculkan dalam angka pada peta sebagai berikut



a. Teman-teman yang diundang, dan rumahnya berjarak lebih dari 17 akan disediakan jemputan. Tentukan teman-teman Patra yang diundang dan akan dijemput.
b. Tuliskan langkah-langkah Anda mendapatkan solusi tersebut dalam bentuk flowchart/pseudocode
c. Dapatkah Anda menggeneralisasi langkah-langkah algoritmik tsb sedemikian hingga dapat digunakan pada peta-peta yang lain?

6. **(UTS 2021)** Robot mini menggambarkan jejak gerakan di atas meja gambar. Untuk melakukannya, dia dapat menjalankan perintah-perintah berikut :

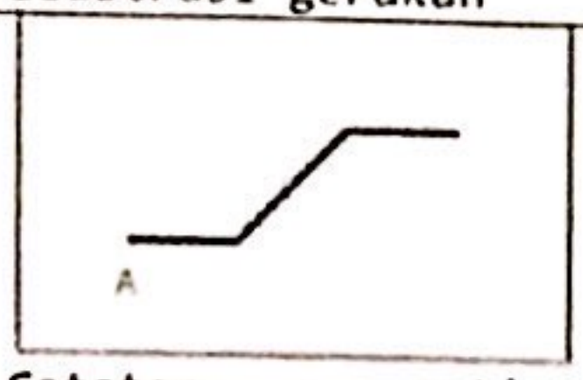
Perintah	Makna
forward	Maju 1 langkah
turn *L/R d*	Berputar/rotasi sebesar d derajat ke kiri jika L/R berniali L, dan ke kanan jika L/R bernilai R. Asumsikan : nilai L/R adalah R atau L, sedangkan 0<=d<=360. Jika d = 0, atau 360, maka robot tidak berubah arah.
repeat *n (perintah)*	Mengulang *perintah* sebanyak *n* kali. Asumsikan n>=0. Jika n=0, maka *perintah* tidak dilakukan sama sekali.
ifelse *(ekspresi) (perintah-1) (perintah-2)*	Jika *ekspresi* bernilai true, maka *perintah-1* dijalankan; Jika ekspresi bernilai false, maka *perintah-2* dijalankan. *Ekspresi* adalah ekspresi boolean seperti yang dapat diajarkan di kelas.



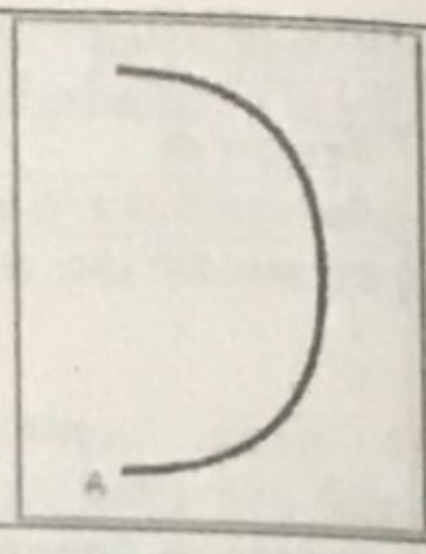
Beberapa informasi tambahan :

- Asumsi : Di awal, robot berada di posisi A dan robot menghadap ke kanan/timur (lihat gambar di atas)
- Robot menerima rangkaian perintah yang dituliskan dalam urutan yang masing-masing dipisahkan sebuah koma (kecuali untuk perintah terakhir, lihat contoh di bawah). Rangkaian perintah dapat dituliskan menyamping, atau dapat juga ditulis ke bawah, dan dapat juga dikombinasikan dengan spasi dan indentasi untuk memperjelas penulisan.
- Parameter *perintah* pada perintah **repeat** dan parameter *perintah-* dan parameter *perintah-2* pada perintah **ifalse** terdiri atas perintah-perintah yang bisa dijalankan oleh robot dan bisa terdiri atas lebih dari 1 perintah.

Contoh :

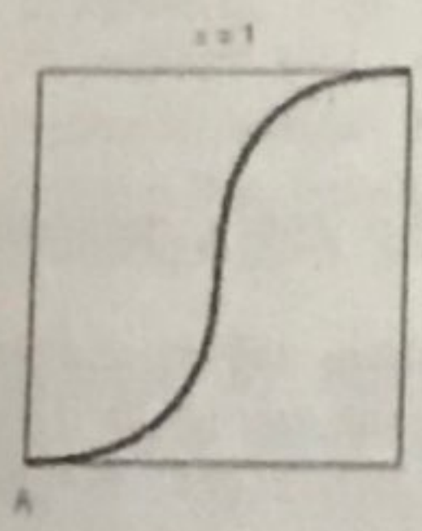
No.	Perintah	Arti	Ilustrasi gerakan
1	**forward, turn L 45, forward, turn R 45** alternatif penulisan : **forward,** **turn L 45,** **forward,** **turn R 45**	Pertama-tama robot maju 1 langkah (ke kanan), lalu berputar ke kiri 45 derajat, lalu maju 1 langkah dan selanjutnya berputar ke kanan 45 derajat.	 Catatan : setiap segmen garis diasumsikan sebesar 1 satuan gerakan

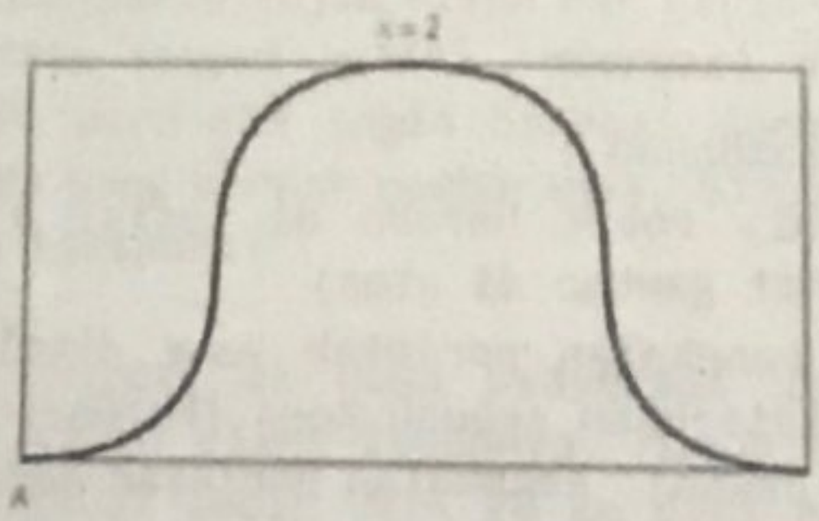
Dipindai dengan CamScanner

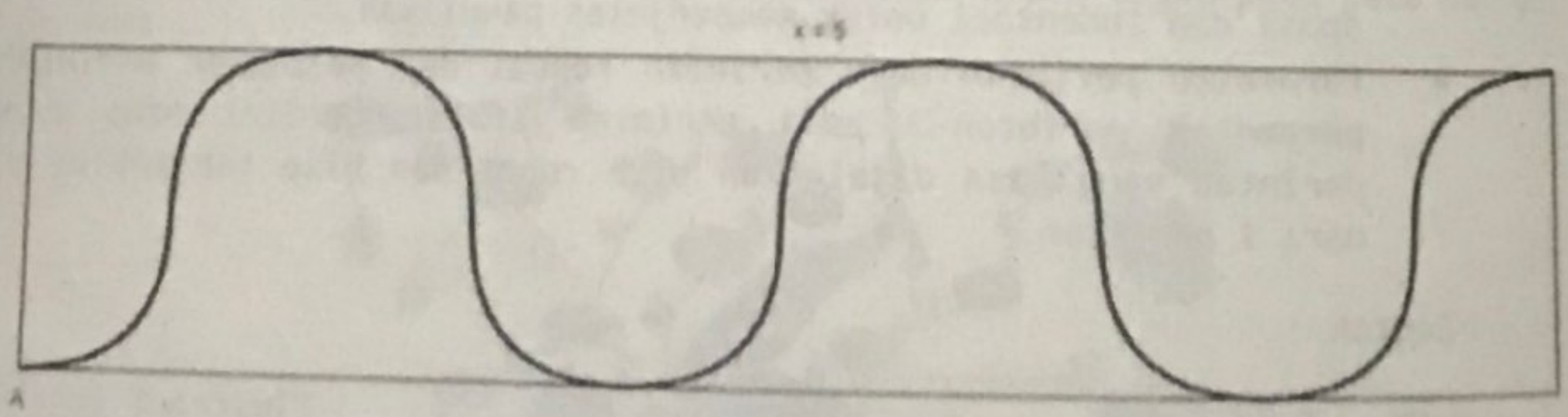
2	**repeat 180 (forward, turn L 1)** Alternatif penulisan: **Repeat 180 (forward, turn L 1)**	Robot mengulang sebanyak 180 kali : maju 1 langkah dan berputar 1 derajat ke kiri. Ini akan membuat robot menggambar setengah lingkaran.	

Tugas Anda :

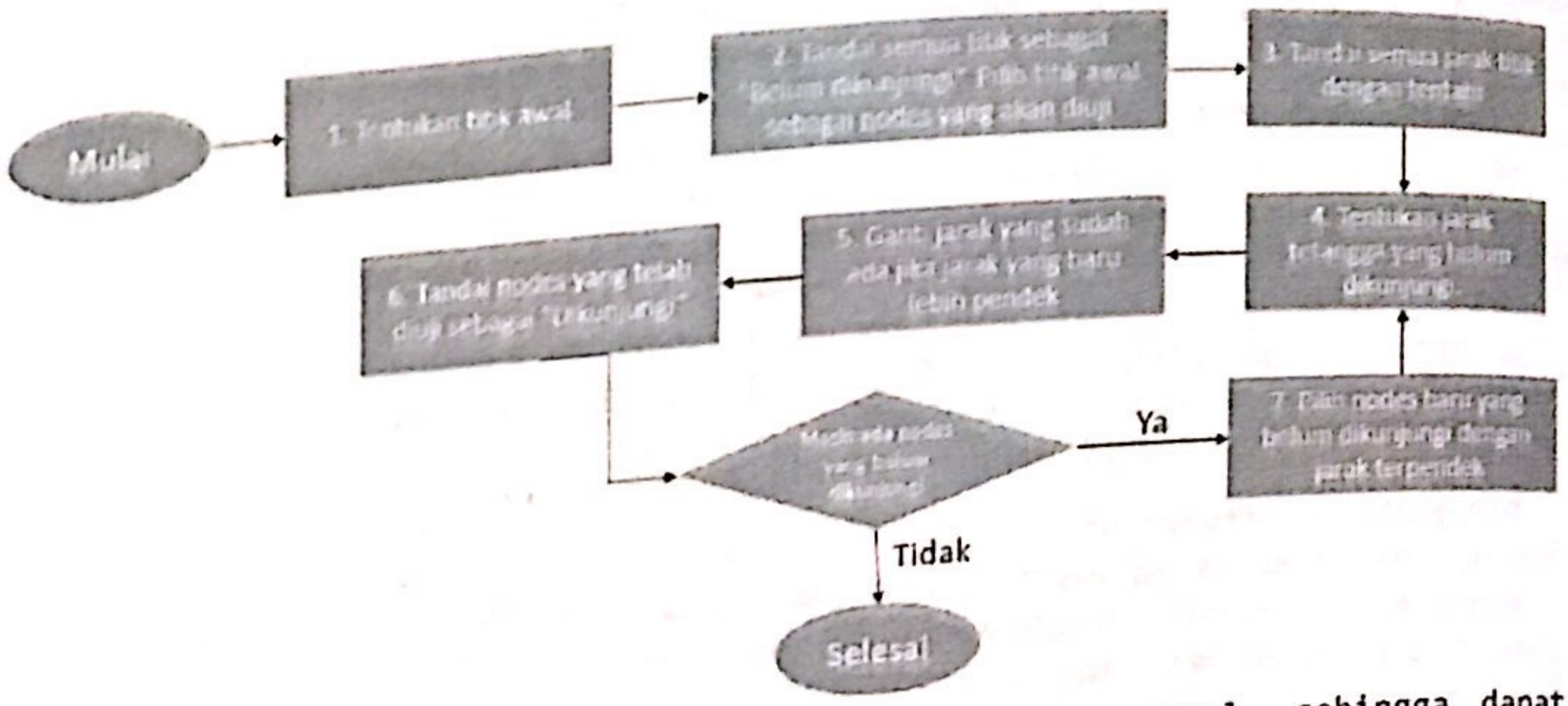
Diberikan **x** (bilangan bulat >=1), tuliskan **urutan perintah** untuk membuat jejak robot dari posisi A ke posisi terakhir untuk membentuk pola huruf S berulang seperti pada contoh-contoh di bawah. Asumsikan selalu ada ruang yang cukup untuk menggambar pola dari posisi manapun. Harap diingat, di bawah ini hanyalah gambar contoh. Silahkan generalisasi, bagaimana pola gambar jika x bernilai lain.







Dipindai dengan CamScanner



Algoritma Dijkstra merupakan algoritma yang sudah general, sehingga dapat digunakan untuk peta apapun. Holt dan Blyth adalah teman Patra yang berjarak <= 20km, sehingga ia diundang. Holt dan Blyth juga berjarak >17km, sehingga ia mendapat jemputan. Berikut adalah hasil dari algoritma dijkstra tersebut :

Exeter	Bath	Ranby	Wells	Mold	Penmon	Saron	Blyth
23	8	12	2	17	24	16	18

Bangor	Stow	Rand	Weir	Cemlyn	Holt	Bilby	Harlow
21	17	13	25	35	19	21	28

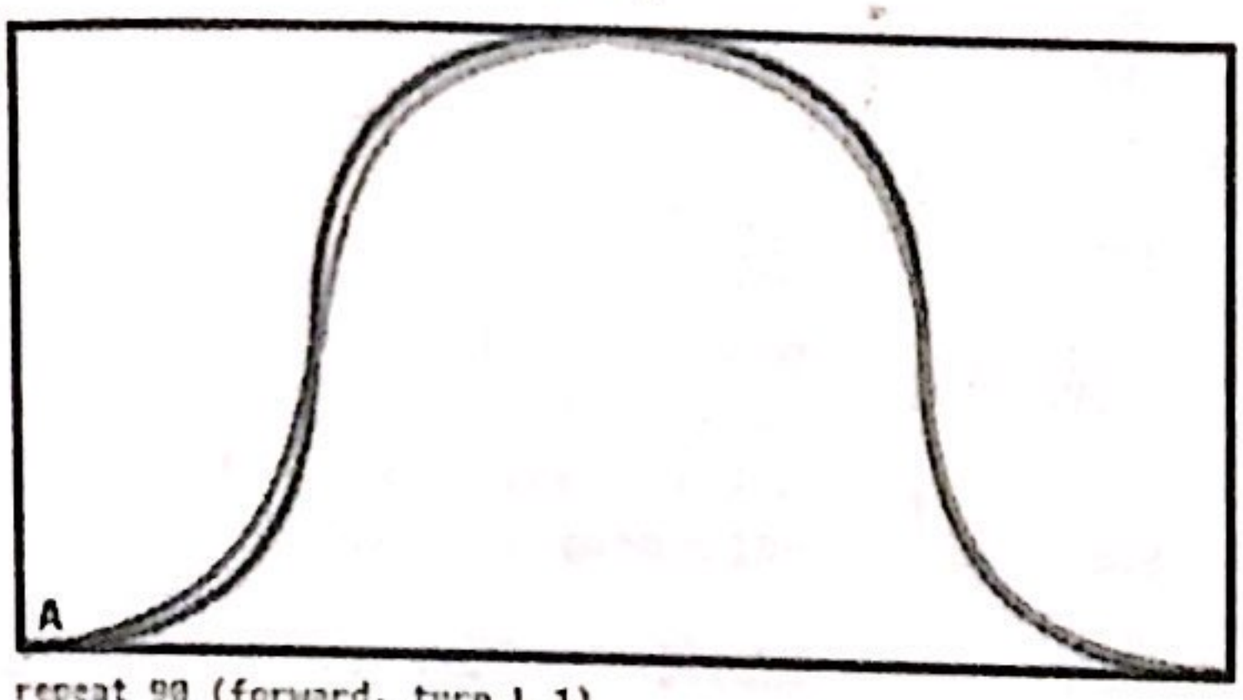
Untuk penjelasan video animasi dapat scan QR Code berikut :

6. (UTS 2021) Berikut jawaban untuk No. 6 :

```
1. i = 1,
2. d = 1,
3. repeat x (
4.  ifelse (i%2==1)
5.          (d = 1)
6.          (d = 359),
7.  repeat 90 (forward, turn L d),
8.  repeat 90 (forward, turn R d),
9.  i+=1
10. )
```

- Pertama-tama, untuk satu kali x, kita perlu memutar robot setengah lingkaran ke kiri. Berdasarkan contoh 2, kita hanya perlu mengubah 180 menjadi 90 agar menjadi setengah lingkaran.
- Selanjutnya kita ingin robot juga membuat setengah lingkaran tapi ke kanan, sehingga L diganti dengan R. Untuk kasus x lebih dari 1, kita harus mengulang program sebanyak x. Maka gunakan program repeat x (...). Namun perlu diperhatikan, untuk x bagian pertama, arahnya adalah kiri-kanan, sedangkan bagian x kedua adalah kanan-kiri. Oleh karena itu, kita perlu modifikasi sedikit, yaitu dengan mengganti nilai derajat menjadi sebuah variabel (n), sehingga untuk mengubah arah tinggal ubah menjadi -1. Modifikasi tersebut dipilih karena mengubah nilai L/R cukup rumit. Perhatikan gambar berikut :

x = 2



Versi berwarna :

```
repeat 90 (forward, turn L 1)
repeat 90 (forward, turn R 1)
repeat 90 (forward, turn R 1) = repeat 90 (forward, turn L 359)
repeat 90 (forward, turn L 1) = repeat 90 (forward, turn R 359)
```

Perhatikan warna pada gambar dan program. Modifikasi nilai derajat dari d menjadi 360-d dapat mengubah arah gerak robot. Misalnya, turn R 1 dan turn L 359 akan menghasilkan arah gerak yang sama. Oleh karena itu, kita perlu variabel pembantu yaitu i sebagai iterasi, dan d sebagai derajat. Untuk setiap x, cek apakah iterasi ini ganjil/genap. Apabila ganjil, maka putaran yang dilakukan adalah kiri-kanan (d=1). Apabila genap, puptaran yang dilakukan adalah kanan-kiri (d=359). Scan QR Code di bawah ini untuk animasi video :

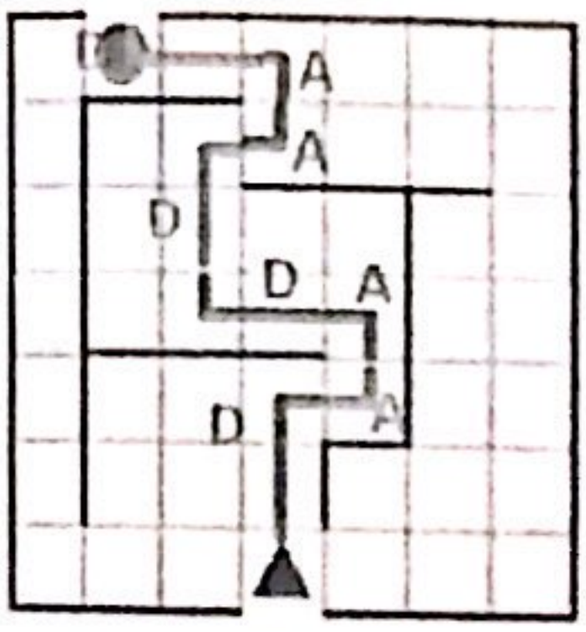
Dipindai dengan CamScanner

PEMBAHASAN

1. Berpikir komputasional adalah cara berpikir yang terinspirasi dari konsep dan prinsip dalam pemrograman dan komputasi. Kita berusaha berpikir untuk memecahkan masalah dengan sistematis, analisis data, abstraksi, dan logis. Berpikir komputasional menyediakan pendekatan sistematis dalam memecahkan masalah. Hal itu sangat penting dalam menghadapi kompleksitas di abad 21 yang semakin kompleks dan beragam. Di era digital ini, kita harus memahami kompleksitas teknologi informasi yang semakin berkembang pesat. Dengan begitu, kita bisa memecahkan masalah yang muncul dalam konteks digital seperti mengolah data, membuat algoritma, berinteraksi dengan teknologi, dan sebagainya.

2. Menggunakan software bajakan adalah ilegal dan haram hukumnya dalam Islam. Salah satu cara agar tetap dapat mengerjakan tugas kita tanpa software ilegal adalah mencari alternatif lain yang gratis. Misalnya Microsoft Office versi web, adalah software alternatif Microsoft Office offline yang berbayar. GIMP merupakan alternatif dari Adobe Photoshop. Blender merupakan software alternatif dari Autodesk 3DS-Max, dan lain sebagainya.

3. Jawaban : D,A,A,D



Scan QR Code berikut untuk penjelasan video :

4. Jawaban : Kim

   Ben meminjamkan bukunya ke Ted. Kemudian Ted memberikannya ke Anna. Anna mengoper bukunya ke Bill. Kemudian Bill meminjamkannya ke Tom. Tom mengembalikan bukunya ke Bill. Bill memberikannya kembali ke Anna. Anna meminjamkannya ke Sara. Sara kemudian mengembalikan bukunya ke Anna. Anna mengembalikannya ke Ted. Ted kemudian memberikan buku itu ke Kim. Scan QR Code berikut untuk penjelasan video :

5. Jawaban : Blyth dan Holt

   Masalah ini dapat diselesaikan dengan algoritma Dijkstra. Flowchart dari algoritma dijkstra adalah sebagai berikut :

Dipindai dengan CamScanner

BAB III : Input, Output, dan Tipe Data

OUTPUT

Output adalah keluaran. Fungsinya adalah mengeluarkan hasil yang telah Anda lakukan dalam program ke console. Untuk menuliskan output, caranya adalah sebagai berikut :

```
1. print("Hello World!")
```

Program di atas akan menuliskan kata "Hello World!". Anda dapat menggantinya dengan mengubah "Hello World!" menjadi kalimat yang Anda inginkan.

INPUT

Input adalah masukan. Fungsinya adalah untuk memasukkan nilai dari user untuk diproses oleh program yang Anda buat. Untuk menuliskan input, caranya adalah sebagai berikut :

```
1. x = input("Masukkan nilai x : ")
```

Variabel x akan bernilai sesuai yang dimasukkan oleh user. Kalimat "Masukkan nilai x : " akan ditampilkan sebagai pertanda bahwa user akan memasukkan nilai untuk variabel x. Anda dapat menggantinya sesuai kebutuhan. Variabel x dapat dikeluarkan dengan :

```
1. print(x)
```

Untuk memasukan teks dan variabel sekaligus dalam keluaran dapat dilakukan :

```
1. print(f"Nilai dari variabel x
adalah {x}.")
```

TIPE DATA

Ada 4 tipe data yang umum dipakai di mata kuliah Pengenalan Komputasi, yaitu :

bool	Bernilai True (benar) / False (Salah)
int	Bernilai bilangan bulat
float	Bernilai bilangan real
string	Kalimat

Contoh :

```
A = True      # boolean (bool)
B = 1222      # int
C = 10.24     # float
D = "abcd"    # string
```

Variabel A,B,C,D di atas dapat dikeluarkan dengan cara print(A), print(B), dan seterusnya, yang akan mengeluarkan nilai sesuai yang ditetapkan, yaitu True, 1222, dan seterusnya.

KONVERSI TIPE DATA

Misalkan x adalah sebuah data,

Konversi ke :	Kode :
String	str(x)
Float	float(x)
Int	int(x)
bool	bool(x)

OPERASI

Operator	Deskripsi	Contoh
+	Penjumlahan	1+2 -> 3
-	Pengurangan	4-2 -> 2
*	Perkalian	2*3 -> 6
**	Perpangkatan	2**3 -> 8
/	Pembagian	14/5 -> 2.8
//	Pembagian (dibulatkan ke bawah)	14//5 -> 2
%	Modulo / sisa bagi	14%5 -> 4

Dipindai dengan CamScanner

SOAL-SOAL

1. (H.1 2021) Tuliskan perintah untuk menampilkan teks "Hello World!" pada layar. Contoh :

Input	Output
	Hello World!

2. Tuliskan perintah untuk meminta input dari user berupa nama (str) dan simpan nilainya pada variabel "nama". Contoh :

Input	Output
Masukkan nama Anda: "Kwek"	

3. Melanjutkan soal No. 3, tuliskan perintah untuk menampilkan isi dari variabel "nama" dengan keterangan : "Nama Anda adalah : (nama)". Contoh :

Input	Output
Masukkan nama Anda: "Kwek"	Nama Anda adalah : Kwek

4. Buat variabel "a" dengan nilai 10 dan variabel "b" dengan nilai 3. Tampilkan hasil dari operasi a + b. Contoh :

Input	Output
	13

5. Buat variabel "c" dan "d" sebagai input dari user. Tampilkan hasil dari operasi c * d. Contoh :

Input	Output
Masukkan nilai c : 2 Masukkan nilai d : 4	8

6. Konversi nilai "123" menjadi tipe data integer dan simpan ke dalam variabel "o". Tampilkan isi dari variabel "o". Contoh :

Input	Output
	123

7. Buatlah perintah untuk meminta input dari user dan simpan nilainya pada variabel x dan y. Tuliskan rata-rata kedua nilai tersebut sebagai output! Contoh :

Input	Output
Masukkan nilai x : 4 Masukkan nilai y : 1	Rata-rata dari x dan y adalah 2.5

8. Misalkan sebuah fungsi berikut :

Ώώ ςώ σ

Terimalah input (int) dari user sebagai x dan keluarkan hasil dari f(x) sebagai output! Contoh :

Input	Output
Masukkan nilai x: 3	Hasil dari f(3) adalah 9

9. Misalkan sebuah fungsi berikut :

Ώώ σώ ςώ φ

Terimalah input (int) dari user sebagai x dan keluarkan hasil dari f(x) sebagai output! Contoh :

Input	Output
Masukkan nilai x: 4	Hasil dari f(4) adalah 62

Dipindai dengan CamScanner

10. Misalkan sebuah mobil bergerak dengan kecepatan v m/s (int) dalam waktu t sekon (int). Buatlah program untuk menentukan jarak yang ditempuh mobil!

Contoh :

Input	Output
Masukkan kecepatan (m/s) = 5 Masukkan waktu (s) = 3	Jarak = 15.0 m

11. Misalkan sebuah fungsi berikut :

$$\text{Ω}\acute{\omega}\tilde{\omega}\dot{\omega} \quad \text{ς}\dot{\omega} \quad \sigma\dot{\omega} \quad \text{τ}\dot{\omega}$$

Terimalah input (float) dari user sebagai x,y, dan z dan keluarkan hasil dari f(x,y,z) sebagai output!

Contoh :

Input	Output
Masukkan nilai x : 1 Masukkan nilai y : 2 Masukkan nilai z : 3	Hasil dari f(1.0,2.0,3.0) adalah 26.0

12. Sebuah persamaan garis lurus seperti berikut :

$$\tilde{\omega}\dot{\omega} \quad \tilde{\omega}\dot{\omega} \quad \tilde{\omega} \quad \pi$$

Buatlah program yang menerima input (int) a,b, dan c, lalu hitunglah gradiennya (m) dan tampilkan hasilnya sebagai output!

Contoh :

Input	Output
Masukkan nilai a : 2 Masukkan nilai b : 5 Masukkan nilai c : 1	Gradien dari 2x+5y+1 adalah -0.4

13. Sebuah toko sepatu menjual 3 jenis sepatu. Berikut adalah pilihan jenis dan harga di toko sepatu tersebut :

Sepatu	Harga
Addidas	Rp1.100.000
Eiger	Rp1.250.000
Nike	Rp1.450.000

Kwak ingin menjadi reseller dan membeli x buah sepatu Addidas, y buah sepatu Eiger, dan z buah sepatu Nike. Tentukan modal yang dibutuhkan Kwak!

Contoh :

Input	Output
Jumlah sepatu Addidas = 2 Jumlah sepatu Eiger = 3 Jumlah sepatu Nike = 1	Total yang harus dibayarkan = 7400000 Rupiah

14. Sebuah termometer menunjukkan suhu sebesar F dengan satuan Fahrenheit. Buatlah program yang menerima nilai F (float) dan mengeluarkan berapa suhu tersebut dalam Celcius!

Contoh :

Input	Output
Masukkan suhu dalam Fahrenheit : 70	Suhu 70.0 F sama dengan 21.111111 C

Dipindai dengan CamScanner

15. Picnometer kosong memiliki berat A gram (float), sedangkan picnometer yang berisi fluida memiliki berat B gram (float). Volume picnometer tersebut adalah 25mL. Tentukan densitas fluida (dalam g/mL) dengan menerima input A dan B, dan mengeluarkan output densitas fluida tersebut! Hint : „ ——

Contoh :

Input	Output
Masukkan berat picnometer kosong (gram) : 12 Masukkan berat picnometer terisi (gram) : 35	Massa jenis fluida tersebut adalah 0.92 g/mL

16. **(UTS 2018)** Buatlah sebuah program untuk menghitung sisi miring sebuah segitiga siku-siku dengan rumus Pythagoras. Program membaca input 2 bilangan riil (float), misalnya a(alas segitiga) dan b (tinggi segitiga) dan menuliskan ke layar hasil perhitungan sisi miring c yang dihitung dengan rumus sbb.: ω Hint : Untuk menghitung x^2, gunakan x**2 dan untuk ω gunakan x**(0.5).

Contoh :

Input	Output
Masukkan alas segitiga : 3 Masukkan tinggi segitiga : 4	Panjang sisi miring adalah 5.0

17. **(UTS 2021)** Jika :

NA = banyak SKS mata kuliah yang mendapat nilai A
NB = banyak SKS mata kuliah yang mendapat nilai B
NC = banyak SKS mata kuliah yang mendapat nilai C
ND = banyak SKS mata kuliah yang mendapat nilai D
NE = banyak SKS mata kuliah yang mendapat nilai E

Maka nilai rata-rata (NR) seorang mahasiswa dalam suatu semester dihitung sebagai berikut:

$$\text{NR} = \frac{\text{NA} \times 4 + \text{NB} \times 3 + \text{NC} \times 2 + \text{ND} \times 1}{\text{NA} + \text{NB} + \text{NC} + \text{ND} + \text{NE}}$$

Contoh :

Input	Output
Masukkan banyak SKS yang mendapat nilai A : 6 Masukkan banyak SKS yang mendapat nilai B : 4 Masukkan banyak SKS yang mendapat nilai C : 3 Masukkan banyak SKS yang mendapat nilai D : 3 Masukkan banyak SKS yang mendapat nilai E : 2	NR = 2.5

Dipindai dengan CamScanner

PEMBAHASAN

1. Untuk menuliskan teks pada layar, gunakan print

```
1. print("Hello World!")
```

2. Untuk menerima masukan dari user, gunakan input

```
1. nama = input("Masukkan nama Anda : ")
```

3. Untuk menuliskan teks dan variabel sekaligus, gunakan print(f"{x}")

```
1. nama = input("Masukkan nama Anda : ")
2. print(f"Nama Anda adalah {nama}.")
```

4. Untuk menetapkan sebuah variabel, gunakan sama dengan (=). Untuk menjumlahkan, gunakan operator penjumlahan (+).

```
1. a = 10
2. b = 3
3. print(a+b)
```

5. Setiap masukan yang diterima dari kode input() adalah string. Untuk melakukan operasi matematika, kita harus konversi tipe data menjadi float atau int.

```
1. c = int(input("Masukkan nilai c : "))
2. d = int(input("Masukkan nilai d : "))
3. # Nilai yang diperoleh dari input() adalah string. Jangan lupa
   konversi ke int agar dapat dilakukan operasi
4. x = c*d
5. print(x)
```

6. Berikut pembahasan untuk nomor 6 :

```
1. x = "123"
2. o = int(x)
3. print(o)
```

7. Ingat bahwa rumus dari rata-rata adalah B Ω ώ ò7ώ ώ ὲ ώΩώ ὸ karena data hanya x dan z, maka rata-ratanya adalah —

```
1. x = int(input("Masukkan nilai x : "))
2. y = int(input("Masukkan nilai y: "))
3. # Nilai yang diperoleh dari input() adalah string. Jangan lupa
   konversi ke int agar dapat dilakukan operasi
4. average = (x+y)/2
5. print(f"Rata-rata dari x dan y adalah {average}")
```

8. Untuk menuliskan rumus fungsi, ada baiknya menambahkan tanda kurung untuk menandai mana yang harus dikerjakan lebih dulu.

```
1. x = int(input("Masukkan nilai x : "))
2. # Jangan lupa konversi ke int agar bisa dilakukan operasi
3. f = (2*x) + 3
4. print(f"Hasil dari f({x}) adalah {f}.")
```


Dipindai dengan CamScanner

9. Berikut pembahasan untuk No. 9 :

```
x = int(input("Masukkan nilai x : "))
# Jangan lupa konversi ke int agar bisa dilakukan operasi
f = (3*(x**2))+(2*x)+6
print(f"Hasil dari f({x}) adalah {f}.")
```

10. Berikut pembahasan untuk No. 10 :

```
v = float(input("Masukakkan kecepatan (m/s) = "))
t = float(input("Masukakkan waktu (s) = "))
s = v*t
print("Jarak = ",s,"m")
```

11. Berikut pembahasan untuk No. 11 :

```
x = float(input("Masukkan nilai x : "))
y = float(input("Masukkan nilai y : "))
z = float(input("Masukkan nilai z : "))
f = (2*x) + (3*(y**2)) + (4*z)
print(f"Hasil dari f({x},{y},{z}) adalah {f}.")
```

12. Berikut pembahasan untuk No. 12 :

```
a = int(input("Masukkan nilai a : "))
b = int(input("Masukkan nilai b : "))
c = int(input("Masukkan nilai c : "))
m = -a/b
print(f"Gradien dari {a}x+{b}y+{c} adalah {m}.")
```

13. Berikut pembahasan untuk No. 13 :

```
#Inisiasi Harga Sepatu
Harga_Addidas = 1100000
Harga_Eiger = 1250000
Harga_Nike = 1450000

#Masukan Input Jumlah Sepatu
Beli_Addidas = int(input("Jumlah Sepatu Addidas = "))
Beli_Eiger = int(input("Jumlah Sepatu Eiger = "))
Beli_Nike = int(input("Jumlah Sepatu Nike = "))

#Hitung total biaya dengan rumus penjumlahan
Total = ((Harga_Addidas*Beli_Addidas) + (Harga_Eiger*Beli_Eiger) +
(Harga_Nike*Beli_Nike))

#Keluarkan hasil berupa luas
print("Total yang harus dibayarkan =", Total,"Rupiah")
```

14. Ingat bahwa rumus Fahrenheit ke Celcius adalah ö Ö σ ς ᶻ -.

```
fahrenheit = float(input("Masukkan suhu dalam Fahrenheit : "))
celcius = (fahrenheit - 32) * 5/9
print(f"Suhu {fahrenheit} F sama dengan {celcius} C")
```


Dipindai dengan CamScanner

15. Berikut pembahasan untuk soal No. 15 :

```
1. A = float(input("Masukkan berat picnometer kosong (gram) : "))
2. B = float(input("Masukkan berat picnometer terisi (gram) : "))
3. V = 25
4. rho = (B-A)/V
5. print(f"Massa jenis fluida tersebut adalah {rho} g/mL")
```

16. Ingat rumus pythagoras bahwa $\dot{\omega}$ $\dot{\omega}$ $\vec{\omega}$ $\vec{\omega}$ 8Gunakan x**n untuk $\dot{\omega}$. Ingat bahwa $\vec{\omega}$ $\dot{\omega}^{8}$. Sehingga, untuk $\vec{\omega}$ dapat menggunakan y**0.5 . Jangan lupa bahwa di Python menggunakan titik sebagai pemisah desimal, bukan koma.

```
1. a = float(input("Masukkan alas segitiga : "))
2. b = float(input("Masukkan tinggi segitiga : "))
3. c = ((a**2)+(b**2))**(0.5)
4. print(f"Panjang sisi miring adalah {c}.")
```

17. Nilai SKS umumnya adalah bilangan bulat. Oleh karena itu, asumsikan nilai SKS selalu bilangan bulat. Kita boleh menggunakan int.

```
1. NA = int(input("Masukkan banyak SKS yang mendapat nilai A : "))
2. NB = int(input("Masukkan banyak SKS yang mendapat nilai B : "))
3. NC = int(input("Masukkan banyak SKS yang mendapat nilai C : "))
4. ND = int(input("Masukkan banyak SKS yang mendapat nilai D : "))
5. NE = int(input("Masukkan banyak SKS yang mendapat nilai E : "))
6. NR = ((NA*4)+(NB*3)+(NC*2)+(ND))/(NA+NB+NC+ND+NE)
7. print(f"NR = {NR}")
```


Dipindai dengan CamScanner

BAB IV : Operator dan Percabangan

Operator dalam Python adalah simbol yang digunakan untuk melakukan operasi pada satu atau beberapa nilai atau variabel.

Operator Assignment

Operator ini digunakan untuk memberikan nilai pada variabel. Operator assignment (=) berfungsi untuk menetapkan suatu variabel. Misalkan X = 5, maka X akan bernilai 5.

Operator	Deskripsi	Cth	Hasil contoh
+=	Penjumlahan	X+=3	8
-=	Pengurangan	X-=3	2
*=	Perkalian	X*=3	15
/=	Pembagian	X/=3	1.667
//=	Pembagian (dibulatkan ke bawah)	X//=3	1
%=	Modulo / sisa bagi	X%=3	2

Operator Relasional

Operator ini digunakan untuk membandingkan dua nilai dan mengembalikan nilai kebenaran (True atau False) tergantung pada apakah pernyataan yang diberikan benar atau salah. Operator ini akan menghasilkan sebuah bool yang bernilai true/false.

Operator	Deskripsi	Contoh True	Contoh False
==	Sama dengan	3 == 3	3 == 2
!=	Tidak sama dengan	1 != 2	1 != 1
<	Kurang dari	2 < 4	2 > 4
>	Lebih dari	4 > 2	4 < 2
<=	Kurang dari sama dengan	5 <= 7	7 <= 5
>=	Lebih dari sama dengan	5 >= 5	5 >= 8

Operator Logika

Operator ini digunakan untuk melakukan operasi logika pada dua atau lebih nilai atau variabel dan mengembalikan nilai kebenaran tergantung pada logika yang digunakan. Operator ini akan menghasilkan sebuah bool yang bernilai true/false

Opertor	Deskripsi	Contoh True	Contoh False
and	Dan (True jika keduanya benar)	(1<2) and (4>3)	(1<2) and (4==3)
or	Atau (True jika salah satunya benar)	(2>2) or (3>2)	(2>2) or (3==2)
not	Tidak / Negasi (True jika salah)	not (2>3)	not (2<3)

Percabangan

Percabangan atau conditional statement dalam Python adalah struktur kontrol alur program yang memungkinkan program untuk memilih aksi yang berbeda tergantung pada kondisi tertentu. Dalam percabangan, program akan mengevaluasi suatu kondisi atau ekspresi boolean (biasanya dengan operator relasional) dan kemudian melakukan tindakan tertentu berdasarkan apakah kondisi tersebut benar atau salah.

- **If statement :**

Dipindai dengan CamScanner

If statement digunakan untuk menjalankan kode jika suatu kondisi terpenuhi. Jika kondisi tersebut benar, program akan mengeksekusi blok kode yang berada di dalam if statement. Jika kondisi tersebut salah, program akan melanjutkan eksekusi pada baris berikutnya di luar if statement. Misalkan x adalah sebuah bool :

```
1. if(x):
2.     ## Program yang dijalankan bila benar
```

Contoh :

```
1. if(A<0):
2.   print("Nilai x kurang dari nol")
```

- **If-else statement :**

If-else statement digunakan untuk menjalankan kode yang berbeda tergantung pada apakah suatu kondisi terpenuhi atau tidak. Jika kondisi tersebut benar, program akan mengeksekusi blok kode yang berada di dalam if. Jika kondisi tersebut salah, program akan mengeksekusi blok kode yang berada di dalam else. Contoh :

```
1. if(x):
2.   ## Program yang dijalankan bila x benar
3. else:
4.   ## Program yang dijalankan bila x salah
```

Contoh :

```
1. if(A<0):
2.   print("Nilai A negatif")
3. else:
4.   print("Nilai A positif atau nol")
```

- **If-elif-else statement :**

If-elif-else statement digunakan untuk menjalankan kode yang jika suatu kondisi terpenuhi, program akan mengeksekusi blok yang ada di dalam if statement. Jika kondisi salah, program akan mengecek kondisi selanjutnya dalam elif. Contoh :

```
1. if(x):
2.   ## Program yang dijalankan bila x benar
3. elif (y):
4.   ## Program yang dijalankan bila x salah tapi y benar
5. else:
6.   ## Program yang dijalankan bila x dan y salah
```

Contoh :

```
1. if(A<0):
2.   print("Nilai A negatif")
3. elif(A>0):
4.   print("Nilai A positif")
5. else:
6.   print("Nilai A = 0")
```

Anda dapat menambahkan elif berapa pun yang Anda mau. Anda juga bisa menghapus else apabila tidak diperlukan.

SOAL-SOAL

Dipindai dengan CamScanner

1. Nilai x dan y (int) adalah input pengguna. Tentukan bilangan terbesar! Contoh :

Input	Output
Masukkan nilai x : 4 Masukkan nilai y : 3	Nilai terbesar adalah 4

2. Air akan membeku pada suhu <0C, mencair pada 0-100 C, dan menguap pada suhu di atas 100C. Pengguna akan memasukkan suhu air (int). Anda diminta untuk menentukan apakah air tersebut berwujud es, cair, atau uap. Contoh :

Input	Output
Masukkan suhu air (C) : 84	Cair

3. Nilai x, y, dan z adalah masukan pengguna. Tentukan urutan bilangan tersebut dari yang terbesar! Contoh :

Input	Output
Masukkan nilai A : 5 Masukkan nilai B : 8 Masukkan nilai C : 2	8,5,2

4. (H.1 2021) Nilai x adalah masukan pengguna. Tentukan apakah nilai x positif, negatif, atau nol. Bila positif, tentukan apakah nilai x adalah ganjil atau genap. Contoh :

Contoh	Input	Output
Contoh 1	Masukkan X: 8	X adalah bilangan positif genap
Contoh 2	Masukkan X: -1	X adalah bilangan negatif
Contoh 3	Masukkan X: 0	X adalah bilangan nol

5. (H.1 2021) Nilai a, b, dan opt adalah masukan pengguna. Variabel a dan b adalah int dan opt adalah operator. Nilai dari opt dapat berupa operator penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan modulo (%). Buatlah program yang menerima a,b, dan opt dan keluarkan hasil dari a (opt) b. Contoh :

Input	Output
Masukkan a : 2 Masukkan b : 3 Masukkan operator : *	2*3=6

6. Kwak memperoleh nilai akhir sebesar x pada mata kuliah pengenalan komputasi. Berikut ini adalah tabel indeks untuk mata kuliah tersebut :

Nilai akhir (x)	Indeks
x > 80	A
75 <= x < 80	AB
70 <= x < 75	B
60 <= x < 70	BC
50 <= x < 60	C
40 <= x < 50	D
x < 40	E

Tentukan indeks Pengenalan Komputasi yang Kwak peroleh!
Contoh :

Input	Output
Masukkan nilai akhir : 62	Indeks untuk nilai 62 adalah BC

7. (UTS 2018) Kelas pertandingan seorang atlit pertandingan bela diri ditentukan berdasarkan berat badannya sebagai beriku t:

Dipindai dengan CamScanner

- τ ὧ ὦ ὠ ὡ ὢ ὣ ὤ ὥ ὦ ὧ

11. (P.1 2021) Setelah Kwuk, kini Kwek melakukan perjalanan dengan pesawat, namun tidak ke luar negeri. Kwek berangkat pukul J_1:M_1:D_1, dan tiba pukul J_2:M_2:D_2. Berapa lama Kwek terbang? (Tuliskan output dalam HH jam MM menit SS detik)
Contoh :

Input	Output
Masukkan waktu keberangkatan Jam : 2 Menit : 15 Detik : 30 Masukkan waktu kedatangan Jam : 5 Menit : 35 Detik : 0	Lama perjalanannya adalah 3 jam 19 menit 30 detik

12. (P.1 2021) Angka favorit Kwok adalah yang memiliki 4 angka, serta jumlah dan hasil kali digit-digitnya dapat membagi angka tersebut. Jika hanya jumlahnya saja yang dapat membagi atau hasil kalinya saja yang dapat membagi, angka tersebut dapat dikategorikan angka cukup favorit. Selain itu, ia menganggap angka itu angka biasa. Kwok akan menginput 4 angka dalam 1 kali input (bukan 4 kali input). Buat program untuk menentukan apakah angka tersebut angka favorit, angka cukup favorit, atau angka biasa. Asumsikan tidak ada angka 0 dalam 4 angka yang diinput.
Contoh :

Contoh	Input	Output	Penjelasan
Contoh 1	1234	Angka biasa	10 dan 24 tidak dapat membagi 1234
Contoh 2	3345	Angka cukup favorit	Hasil penjumlahan (15) dapat membagi 3345, tetapi hasil kali (180) tidak dapat membagi 3345.
Contoh 3	1116	Angka favorit	Hasil penjumlahan (9) dan hasil kali (6) dapat membagi 1116.

13. Sebuah persamaan kuadrat adalah sebagai berikut :

$$\text{[illegible]} \quad \text{[illegible]} \quad \pi$$

Buatlah program untuk mencari akar-akar dari persamaan tersebut dengan ketentuan berikut :

- Jika a=0, tuliskan "persamaan tidak valid"
- Nilai diskriminan adalah Ω [illegible]
  - Jika d < 0, maka tuliskan "Angka tidak memiliki akar real"
  - Jika d = 0, maka hitung akar tunggal ([illegible]
  - Jika d > 0, maka hitung dua nilai akar tersebut ([illegible]

Contoh :

Contoh	Input	Output
Contoh 1	Masukkan nilai a: 2 Masukkan nilai b: 1 Masukkan nilai c: 3	Persamaan tidak memiliki akar real
Contoh 2	Masukkan nilai a: 1 Masukkan nilai b: -2 Masukkan nilai c: 1	Persamaan memiliki satu akar tunggal : x = 1.0
Contoh 3	Masukkan nilai a: 1 Masukkan nilai b: 2	Persamaan memiliki dua akar: x1=1.0 dan x2 = -3.0

Dipindai dengan CamScanner

Masukkan nilai c: -3

14. Kweak adalah seorang penjual kopi. Bahan-bahan yang ia gunakan adalah biji kopi, susu, dan gula dengan perbandingan A:B:C. Kweak ingin salah satu bahan harus habis digunakan. Suatu hari ia belanja sebanyak x gram biji kopi, y gram susu, dan z gram gula. Buatlah program untuk menentukan berapa banyak maksimal kopi yang dapat ia jual pada hari itu!

Contoh :

Input	Output
Tentukan Nilai A: 2 Tentukan Nilai B: 4 Tentukan Nilai C: 2 Banyaknya biji kopi: 21 Banyaknya susu: 28 Banyaknya gula: 18	Banyak kopi yang dapat dibuat sebanyak 7.

15. **(UTS 2021)** Salah satu sistem yang biasa digunakan untuk menentukan golongan darah manusia adalah sistem ABO. Menurut sistem ABO, golongan darah manusia dikontrol oleh 3 macam alel, yaitu alel IA, IB, dan IO. Seseorang mewarisi satu alel dari ayah dan satu alel dari ibu dan kombinasi dari alel warisan dari ayah dan ibu menentukan golongan darahnya. Tabel berikut berisi kriteria pembentukan alel ganda pada anak berdasarkan yang diwarisi dari kedua orang tuanya dan golongan darah anak berdasarkan alel gandanya.

Orang Tua-1	Orang Tua-2	Anak	
		Alel Ganda	Golongan Darah
IA	IA	IAIA	A
	IB	IAIB	AB
	IO	IAIO	A
IB	IB	IBIB	B
	IO	IBIO	B
IO	IO	IOIO	O

Buatlah program yang menerima masukan alel yang diwarisi dari ibu, misalnya Ix, dan alel yang diwarisi dari ayah, misalnya Iy, dan menuliskan ke layar golongan darah anak berdasarkan informasi di atas. Asumsikan masukan alel selalu benar seperti ketentuan di atas.

Contoh :

Contoh	Input	Output
Contoh 1	Masukkan alel yang diwarisi ibu (Ix) : IB Masukkan alel yang diwarisi ayah (Iy) : IA	Golongan darah Anda adalah AB
Contoh 2	Masukkan alel yang diwarisi ibu (Ix): IA Masukkan alel yang diwarisi ayah (Iy): IO	Golongan darah Anda adalah A

Dipindai dengan CamScanner

PEMBAHASAN

1. Untuk mencari nilai terbesar, buat asumsi nilai terbesar terlebih dahulu, lalu bandingkan hasil asumsi dengan nilai lainnya. Bila nilai lainnya lebih besar, maka catat nilai tersebut sebagai nilai terbesar.

```
1. # Masukkan nilai x dan y
2. x = int(input("Masukkan nilai x : "))
3. y = int(input("Masukkan nilai y : "))
4.
5. # Asumsikan yang terbesar adalah x
6. terbesar = x
7.
8. # Cek apakah y lebih besar dari (terbesar)
9. if(y>terbesar):
10.     terbesar = y
11. print(f"Nilai terbesar adalah {terbesar}")
```

2. Untuk membandingkan nilai yang memiliki range, mulailah dengan mengecek pada range terbesar. Setelah itu, lanjutkan dengan elif. Elif ini akan mengeliminasi pernyataan if/elif sebelumnya. Misalkan :

```
1. if (suhu>80):
2.   print("Suhu lebih dari 80")
3. elif (suhu>20):
4.   print("Suhu kurang dari 80, tetapi lebih dari 20")
5. else:
6.   print("Suhu kurang dari 20")
```

Perhatikan pada line 3, elif tersebut mengeliminasi pernyataan if/elif sebelumnya yaitu suhu>80. Print pada line 4 tersebut hanya akan keluar jika pernyataan suhu>80 salah, dan suhu>80 benar. Atau singkatnya, print pada line 4 hanya keluar jika 80>suhu>20.

Untuk soal ini, alur berpikirnya adalah sebagai berikut. Cek apakah suhu lebih dari 100. Jika tidak (elif), cek apakah suhu lebih dari 0. Pengecekan suhu lebih dari 0 setelah pengecekan suhu lebih dari 100 sama saja dengan pengecekan apakah suhu ada di antara 0 hingga 100. Sehingga, solusinya adalah sebagai berikut :

```
1. # Masukkan input
2. suhu = int(input("Masukkan suhu air (C) : "))
3.
4. # Cek mulai dari range terbesar
5. if(suhu>100): # Cek apakah suhu lebih dari 100
6.     print("Uap")
7. elif (suhu>0): # Jika kurang dari 100, maka cek apakah suhu lebih dari 0
8.     print("Cair")
9. else : # Jika kurang dari 100 dan kurang dari 0, sudah pasti beku
10.     print("Es")
```

3. Metode yang digunakan pada soal ini adalah brute-force, yaitu mengecek semua kemungkinan. Karena terdapat 3 nilai, maka semua kemungkinan urutannya adalah 3! = 6 kemungkinan, yaitu : ABC, ACB, BAC, BCA, CAB dan CBA.

Dipindai dengan CamScanner

```
1. # Masukkan input yang dibutuhkan
2. A = int(input("Masukkan nilai A = "))
3. B = int(input("Masukkan nilai B = "))
4. C = int(input("Masukkan nilai C = "))
5.
6. # Cek semua kemungkinan
7. if (A>=B>=C):
8.     print(f"{A},{B},{C}")
9. elif (A>=C>=B):
10.     print(f"{A},{C},{B}")
11. elif (B>=A>=C):
12.     print(f"{B},{A},{C}")
13. elif (B>=C>=A):
14.     print(f"{B},{C},{A}")
15. elif (C>=A>=B):
16.     print(f"{C},{A},{B}")
17. else: # Sudah pasti C>=B>=A
18.     print(f"{C},{B},{A}")
```

4. Untuk mengecek apakah bilangan ganjil atau genap, kita tinggal memodulokan dengan 2. Jika hasilnya adalah 1, maka ia ganjil. Jika hasilnya 0, maka genap. Contoh : 7%2 hasilnya 1. Oleh karena itu, 7 adalah ganjil.

```
1. # Masukkan inputan
2. x=int(input("Masukkan X: "))
3.
4. if(x>0): # Cek apakah nilai x lebih dari 0
5.     if(x%2==0): # Cek apakah sisa bagi dari x/2 adalah 0
6.         print("X adalah bilangan positif genap")
7.         # Jika sisa baginya 0, maka genap
8.     else:
9.         print("X adalah bilangan positif ganjil")
10.         # Jika sisa baginya 1, maka ganjil
11. elif(x<0): # Cek apakah x kurang dari 0
12.     print("X adalah bilangan negatif")
13. else: # Jika bukan lebih dari 0 dan kurang dari 0, maka pasti bilangan 0.
14.     print("X adalah bilangan nol")
```

5. Untuk mengecek operator yang dimasukkan user, kita akan menggunakan string dan membandingkannya satu per satu dalam if-else.

```
1. # Masukkan inputan yang dibutuhkan
2. a = int(input("Masukkan angka pertama: "))
3. b = int(input("Masukkan angka kedua: "))
4. opt = input("Masukkan operator: ")
5.
6. # Cek untuk setiap operator
7. if (opt=="+"):
8.     print(f"{a}+{b}={a+b}")
9. elif (opt=="-"):
10.     print(f"{a}-{b}={a-b}")
11. elif (opt=="*"):
12.     print(f"{a}*{b}={a*b}")
```


Dipindai dengan CamScanner

```
elif (opt=="/"):
    print(f"{a}/{b}={a/b}")
elif (opt=="%"):
    print(f"{a}%{b}={a%b}")
else:
    print("Operator tidak dikenali.")
```

6. Alur berpikir soal ini mirip seperti nomor 2.

```
# Masukkan inputan
x = int(input("Masukkan nilai akhir : "))

# Cek kemungkinan :
if (x>80): # Bila indeks lebih dari 80
    indeks = "A"
elif (x>75): # Bila indeks kurang dari 80, cek apakah lebih dari 75
    indeks = "AB"
elif (x>70): # Bila indeks kurang dari 75, cek apakah lebih dari 70
    indeks = "B"
elif (x>60): # Bila indeks kurang dari 70, cek apakah lebih dari 60
    indeks = "BC"
elif (x>50): # Bila indeks kurang dari 60, cek apakah lebih dari 50
    indeks = "C"
elif (x>40): # Bila indeks kurang dari 50, cek apakah lebih dari 40
    indeks = "D"
else: # Jika di bawah 40
    indeks = "E"
print(f"Indeks untuk nilai {x} adalah {indeks}")

```

7. Konsep berpikir soal ini sama dengan nomor sebelumnya. Anda dapat melakukan hal yang sama seperti sebelumnya. Namun pada kode ini akan dicontohkan penggunaan elif tanpa eliminasi seperti yang dijelaskan nomor 2.

```
# Masukan berat badan
berat_badan = int(input("Masukkan berat badan atlit (kg): "))
# Cek kemungkinan
if berat_badan < 45: # Jika dibawah 45
    print("Tidak memenuhi kualifikasi")
elif 45 <= berat_badan <= 50: # Jika 45<=berat<=50
    print("Kelas A")
elif 50 < berat_badan <= 55: # Jika 50<berat<=55
    print("Kelas B")
elif 55 < berat_badan <= 60: # Jika 55<berat<=60
    print("Kelas C")
elif 60 < berat_badan <= 65: # Jika 60<berat<=65
    print("Kelas D")
elif 65 < berat_badan <= 70: # Jika 65<berat<=70
    print("Kelas E")
else: #Jika di atas 70
    print("Kelas F")
```


Dipindai dengan CamScanner

8. Ingat bahwasanya gradien dari persamaan ώ ά ώ ώ adalah m. Sehingga untuk ώ ώ ώ ώ ώ π, maka gradiennya adalah ά –. Kedua garis sejajar bila $m_1 = m_2$. Kedua garis tegak lurus bila $m_1.m_2=-1$. Bila tidak keduanya, maka kedua garis tidak sejajar dan tidak tegak lurus.

```
1. # Masukkan semua inputan yang dibutuhkan
2. a1 = int(input("Masukkan a1: "))
3. b1 = int(input("Masukkan b1: "))
4. c1 = int(input("Masukkan c1: "))
5. a2 = int(input("Masukkan a2: "))
6. b2 = int(input("Masukkan b2: "))
7. c2 = int(input("Masukkan c2: "))
8.
9. # Hitung gradien masing-masing persamaan
10. m1 = -a1 / b1
11. m2 = -a2/b2
12.
13. # Periksa semua kemungkinan
14. if (m1==m2):
15.     print("Kedua garis sejajar")
16.     # Garis sejajar bila m1=m2
17. elif (m1*m2==-1):
18.     print("Kedua garis tegak lurus.")
19.     # Garis tegak lurus bila m1.m2 = -1
20. else:
21.     # Selain itu maka tidak sejajar dan tidak tegak lurus
22.     print("Kedua garis tidak sejajar dan tidak tegak lurus.")
```

9. Konsep berpikir soal ini mirip dengan soal No.5

```
1. # Masukkan inputan F (int) dan Kode (str)
2. F = int(input("Masukkan suhu dalam Farenheit: "))
3. Kode = input("Masukkan kode suhu: ")
4.
5. # Cek kode
6. if (Kode=="C"):
7.     hasil = int((F-32)*5/9) # F ke C
8.     print("Suhu dalam Celcius adalah",hasil,"C")
9. elif (Kode=="K"):
10.     hasil = int(((F-32)*5/9)+273) # F ke K
11.     print("Suhu dalam Kelvin adalah",hasil,"K")
12. elif (Kode=="R"):
13.     hasil =int(((F-32)*4/9)) # F ke R
14.     print("Suhu dalam Reamur adalah",hasil,"R")
15. else:
16.     # Jika kode bukan F,K, atau R
17.     print("Kode tidak valid.")
18.
```

10. Untuk tipe soal jam seperti ini, kita perlu menyamakan satuan terlebih dahulu. Cara yang paling mudah adalah menjadikan semua satuan (jam, menit, detik) menjadi satuan terkecil (detik). Pada kasus ini, hanya ada jam dan menit. Sehingga

Dipindai dengan CamScanner

satuan terkecilnya adalah menit. Maka konversi semua jam menjadi menit. Setelah itu, baru kita bisa lakukan operasi matematika.

```
1. # Masukkan seluruh inputan
2. j = int(input("Masukkan jam kepergian: "))
3. m = int(input("Masukkan menit kepergian: "))
4. x = int(input("Masukkan lama penerbangan (menit):  "))
5. y = int(input("Masukkan perbedaan waktu (menit): "))
6.
7. # Konversi jam-menit pergi menjadi satuan terkecil (menit).
8. # Misal pergi jam 01:20 akan menjadi 80mnt
9. menit = j*60 + m
10.
11. # Jumlahkan dengan lama penerbangan dan perbedaan waktu
12. menit = menit + x + y
13.
14. # Konversi menit ke jam
15. # Tanda // artinya dibagi dan dibulatkan ke bawah
16. jamdatang = menit//60
17.
18. # Menitnya berarti sisa dari pembagian menit/60
19. menitdatang = menit%60
20.
21. print(f"Waktu datang = jam {jamdatang} lewat {menitdatang} menit")
```

11. Soal ini mirip dengan soal No.10. Pada soal ini satuan terkecilnya adalah detik. Sehingga jam dan menit harus dikonversi menjadi detik agar bisa dilakukan operasi matematika.

```
1. # Masukkan inputan
2. # Tanda \t artinya tab, untuk merapikan output
3. print("Masukkan waktu keberangkatan")
4. J1 = int(input("Jam\t: "))
5. M1 = int(input("Menit\t: "))
6. D1 = int(input("Detik\t: "))
7. print("Masukkan waktu kedatangan")
8. J2 = int(input("Jam\t: "))
9. M2 = int(input("Menit\t: "))
10. D2 = int(input("Detik\t: "))
11.
12. # Konversi jam-menit-detik menjadi satuan terkecil (detik)
13. detik1 = J1*3600 + M1*60 + D1
14. detik2 = J2*3600 + M2*60 + D2
15.
16. # Hitung selisih detik
17. detik3 = detik2-detik1
18.
19. # Konversi detik menjadi jam
20. J3 = detik3//3600
21.
22. # Sebagian detik sudah menjadi jam. Sehingga sisa detik adalah sisa bagi
    dari detik/3600
23. detik3 = detik3%3600
24.
```


Dipindai dengan CamScanner

```
# Konversi detik menjadi menit
M3 = detik3//60

# Sebagian detik sudah menjadi menit. Sehingga sisa detik adalah sisa
bagi dari detik/60
detik3 =detik3%60

# Keluarkan hasil
print(f"Lama perjalanannya adalah {J3} jam {M3} menit {detik3} detik")
```

12. Kita perlu mengetahui angka satuan, puluhan, ratusan dan ribuan.
    - Untuk memperoleh angka satuan (satuan terendah), kita hanya perlu memodulokan dengan 10. Misal 1234%10, maka hasilnya adalah 4.
    - Untuk memperoleh angka ribuan (satuan tertinggi), kita hanya perlu membaginya dengan 1000 (satuan angka tertinggi) lalu bulatkan ke bawah. Misalkan 1234/1000 adalah 1.234, dan dibulatkan ke bawah menjadi 1.
    - Untuk memperoleh angka lainnya, misalnya puluhan dan ratusan, kita perlu memodulo terlebih dahulu dengan satuan di atasnya. Setelah itu kita bagi dengan satuannya dan bulatkan ke bawah. Jika kita ingin memperoleh angka ratusan, maka modulo dulu dengan 1000, lalu bagi dengan 100 dan bulatkan ke bawah. Misalkan 4231 dimodulo 1000 akan menghasilkan 231. Ketika dibagi 100, maka hasilnya 2.31 dan dibulatkan ke bawah menjadi 2.

```
# Masukkan inputan
angka = int(input("Masukkan angka 4 digit: "))

# Pisahkan satuan, puluhan, ratusan dan ribuan
satuan = angka%10 # Sisa bagi 10 akan menghasilkan satuan
ribuan = angka//1000 # Operator // akan menghasilkan pembagian dibulatkan
ke bawah
puluhan = (angka%100)//10 # Sisa bagi angka dengan 100 akan menghasilkan
angka puluhan.
ratusan = (angka%1000)//100 # Sisa bagi angka dengan 1000 akan
menghasilkan angka ratusan.

# Hitung hasil kali dan hasil jumlah
hasiljumlah = ribuan+ratusan+puluhan+satuan
hasilkali = ribuan*ratusan*puluhan*satuan

# Cek kemungkian-kemungkinan
if (angka%hasilkali==0 and angka%hasiljumlah==0):
    print("Angka favorit")
elif (angka%hasilkali==0 or angka%hasiljumlah==0):
    print("Angka cukup favorit")
else:
    print("Angka biasa")
```

13. Berikut pembahasan untuk No. 12 :

```
# Masukkan setiap input
a = float(input("Masukkan nilai a: "))
```


Dipindai dengan CamScanner

```
b = float(input("Masukkan nilai b: "))
c = float(input("Masukkan nilai c: "))

# Memeriksa apakah a sama dengan 0
if a == 0:
    print("Persamaan tidak valid.")
else:
    # Menghitung diskriminan
    diskriminan = b**2 - 4*a*c

    # Memeriksa nilai diskriminan
    if diskriminan < 0:
        print("Persamaan tidak memiliki akar real.")
    elif diskriminan == 0:
        # Menghitung nilai akar tunggal
        x = -b / (2*a)
        print(f"Persamaan memiliki satu akar tunggal: x = {x}")
    else:
        # Menghitung dua nilai akar
        x1 = (-b + (diskriminan**0.5)) / (2*a)
        x2 = (-b - (diskriminan**0.5)) / (2*a)
        print(f"Persamaan memiliki dua akar: x1 ={x1} dan x2 = {x2}")

```

14. Pertama-tama kita harus mengetahui berapa kopi yang dapat dibuat oleh masing-masing bahan tersebut. Misalkan komposisi bahan biji kopi:susu:gula adalah 2:5:1. Maka jika stok biji kopi:susu:gula adalah 21:23:20, maka biji kopi akan habis untuk membuat (21//2=10) gelas kopi; susu akan habis untuk membuat (23//5=4) gelas kopi; dan gula akan habis untuk membuat (20//1=20) gelas kopi. Oleh karena itu, karena susu habis setelah 4 gelas kopi dibuat, maka Firzha tidak dapat membuat kopi lagi. Sehingga kopi yang dapat ia buat adalah 4 gelas (yang minimum).

```
# Masukkan semua inputan
A = int(input("Tentukan Nilai A: "))
B = int(input("Tentukan Nilai B: "))
C = int(input("Tentukan Nilai C: "))
x = int(input("Banyaknya kopi: "))
y = int(input("Banyaknya susu: "))
z = int(input("Banyaknya gula: "))

# Tentukan berapa banyak kopi yang dapat dibuat agar suatu bahan habis
BS1 = x//A
BS2 = y//B
BS3 = z//C

# Tentukan nilai minimum
min = BS1
if (min>BS2):
    min = BS2
if (min>BS3):
    min = BS3
```


Dipindai dengan CamScanner

```
20.
21. # Tuliskan hasilnya
22. print(f"Banyak kopi yang dapat dibuat sebanyak {min}")
```

15. Kita mulai dengan memasukkan input yang kita butuhkan, yaitu alel Ix (ibu) dan Iy (ayah). Program mengecek 2 kemungkinan kombinasi karena pada manusia, alel gen dari setiap orang terdapat dalam pasangan. Dalam hal ini, setiap individu mewarisi alel dari kedua orang tuanya. Oleh karena itu, terdapat kemungkinan bahwa alel yang diwarisi dari ibu dan ayah dapat berada pada posisi yang berbeda pada pasangan alel.
Sebagai contoh, jika seseorang mewarisi alel IA dari ibu dan alel IO dari ayah, maka pasangan alelnya adalah IAIO untuk kombinasi IxIy. Namun, jika seseorang mewarisi alel IO dari ibu dan alel IA dari ayah, maka pasangan alelnya adalah IOIA untuk kombinasi IxIy. Ini tentunya tidak terdapat pada tabel yang diberikan. Kita perlu kombinasi berbeda yaitu IyIx untuk memperoleh IAIO. Oleh karena itu, untuk memastikan bahwa semua kemungkinan kombinasi alel telah dipertimbangkan, program harus memeriksa kedua kemungkinan kombinasi tersebut. Barulah kita bisa mengecek untuk setiap kemungkinan yang ada pada tabel terhadap dua kemungkinan kombinasi tersebut.

```
1. # Masukkan inputan
2. Ix = input("Masukkan alel yang diwarisi ibu (Ix): ")
3. Iy = input("Masukkan alel yang diwarisi ayah (Iy): ")
4.
5. # Ada 2 kemungkinan, yaitu IxIy atau IyIx.
6. IxIy = Ix+Iy
7. IyIx = Iy+Ix
8.
9. # Buat teks yang akan menampilkan hasil
10. teks = "Golongan darah Anda adalah "
11.
12. # Bandingkan semua kemungkinan
13. if ((IxIy=="IAIA" or IyIx == "IAIA") or (IxIy=="IAIO" or IyIx=="IAIO")):
14.     print(teks + "A")
15. elif (IxIy=="IAIB" or IyIx == "IAIB"):
16.     print(teks + "AB")
17. elif ((IxIy=="IBIB" or IyIx == "IBIB") or (IxIy=="IBIO" or
    IyIx=="IBIO")):
18.     print(teks+"B")
19. elif(IxIy=="IOIO" or IyIx == "IOIO"):
20.     print(teks+"O")
21. else:
22.     print("Hasil tidak ditemukan.")
23.
```


Dipindai dengan CamScanner

BAB V : Pengulangan

PENGULANGAN

Pengulangan adalah program yang kita ulang terus menerus untuk mencapai hasil tertentu.

FOR LOOP

For loop adalah pengulangan untuk setiap i dalam variabel. Penulisannya adalah sebagai berikut :

```
1. for i in range(a,n,b):
2.         (Program yang diulang)
```

Program di atas akan mengulang (program yang diulang) untuk i = a, hingga i = n-1 dengan beda b. Misalkan untuk menuliskan angka genap dari 1 sampai 10, maka :

```
1. for i in range(2,11,2):
2.         print(i)
```

Program di atas akan menuliskan 2 4 6 8 10. Perhatikan bahwa n ditambah 1. Program akan mengulang hingga i=n-1. Agar 10 terbawa, maka ditambah satu. Apabila b=1, kita bisa saja tidak menuliskan parameter b. Misalkan program untuk menulis angka 1-10 :

```
1. for i in range(1,11):
2.         print(i)
```

WHILE LOOPS

While loops adalah pengulangan selama suatu kondisi benar. Penulisannya adalah sebagai berikut :

```
1. while (kondisi):
2.         (Program yang diulang)
```

Kondisi harus berupa bool. Misalkan untuk menulis 1-5 menggunakan while :

```
1. num = 1
2. while num <= 5:
3.         print(num)
4.         num += 1
```

Perhatikan kode tersebut. Selama num<=5 benar, maka print(num) dan num+=1 akan terus diulang.

BREAK

Break berfungsi untuk mematahkan pengulangan. Misalkan :

```
1. for i in range(1,10):
2.         print(i)
3.         if(i==7):
4.                 break
```

Program di atas rencananya akan menuliskan i dalam rentang 1 sampai 9. Tetapi jika i=7, patahkan pengulangannya. Sehingga yang tertulis akan hanya 1 2 3 4 5 6 7.

CONTINUE

Continue berfungsi untuk melanjutkan pengulangan dari awal dan mengabaikan program selanjutnya. Misalnya :

```
1. for i in range(1,10):
2.         if(i==7):
3.                 continue
4.         print(i)
```

Program di atas rencananya akan menuliskan i dari 1 sampai 9. Tetapi saat i=7, program akan lanjut untuk i=8 dan mengabaikan program di bawahnya yaitu print(i), sehingga yang tertulis adalah 1 2 3 4 5 6 8 9.

Pengulangan Bersarang

Pengulangan bersarang adalah pengulangan dalam pengulangan. Misalkan sebagai berikut :

```
1. n = 3
2. for i in range(n):
3.         for j in range(i+1):
4.                 print("*", end=" ")
5.         print()
```

Program itu akan menghasilkan :

```
*
**
***
```


Dipindai dengan CamScanner

SOAL-SOAL

1. **(H.2 2021)** Buatlah program untuk menuliskan angka dari 1 - n

   Contoh :

Input	Output
Masukkan n : 4	1 2 3 4

2. **(H.2 2021)** Buatlah program yang menerima bilangan N dan menulis bilangan 10^x terkecil yang lebih dari N. Contoh :

Input	Output
123	1000

3. Buatlah program untuk menghitung jumlah dari angka genap antara 1 sampai N

   Contoh :

Input	Output
Masukkan N :	650

4. **(H.2 2021)** Buatlah program yang menerima bilangan N, dan cek apakah N bilangan prima

   Contoh :

Input	Output
Masukkan N : 5	Prima

5. Buatlah program untuk menerima bilangan N, dan menjumlahkan semua bilangan kelipatan 5 antara 1 sampai N.

   Contoh :

Input	Output
Masukkan N : 128	1625

6. **(UTS 2021)** Buatlah program untuk menerima nilai a,b, dan x (bilangan real) dan n (bilangan bulat), lalu hitung persamaan berikut :

   Ωώ ώώ ώ

   Contoh :

Input	Output
Masukkan a: 2 Masukkan b: 5 Masukkan x: 3 Masukkan n : 4	260.0

7. **(UTS 2021)** Bilangan euler (e) diperoleh dari deret berikut :

   Ω ρ ρ/ρΑ ρ/ςΑ ρ/σΑ8

   Buatlah program untuk menentukan bilangan euler dengan menghitung deret di atas hingga suku ke-N.

   Contoh :

Input	Output
Masukkan nilai n: 10	2.718281801146384

8. **(UTS 2021)** Diketahui perkiraan jumlah penduduk di kota A pada tahun ke-t adalah :

   ΰ ό ΰ Ω

   Dengan P_0 = 1,2 juta, dan k = 0.015. Buat program yang menerima bilangan m, dan tentukan perkiraan populasi tahun ke 1 sampai m. Gunakan hampiran euler dengan deret pada nomor 7 dengan 100 suku. Contoh :

Input	Output
Masukkan m:	Tahun ke-1: 1.2 juta Tahun ke-2: 1.218... juta Tahun ke-3: 1.236... juta

9. Buatlah program untuk menerima bilangan bulat N dan menuliskan faktor-faktornya. Faktor-faktor suatu bilangan bulat N adalah bilangan-bilangan bulat positif yang habis membagi N.

   Contoh :

Input	Output
Masukkan N: 15	15 5 3

Dipindai dengan CamScanner

10. Buatlah program yang menerima bilangan N. N selanjutnya adalah indeks mahasiswa. Indeks yang lulus adalah A,B,C, dan D. Dan yang tidak lulus adalah E. Program mengeluarkan berapa banyak yang lulus dan tidak lulus. Contoh :

Input	Output
Masukkan jumlah mahasiswa: 5 A E B	Lulus : 2 Tidak lulus : 1

11. Buatlah program untuk membaca sekumpulan bilangan bulat positif, dan menghitung berapa banyak bilangan genap dan ganjil. Program berhenti menerima data saat memasukkan angka negatif.
Contoh :

Input	Output
Masukkan angka: 12 Masukkan angka: 4 Masukkan angka: 13 Masukkan angka: 94 Masukkan angka: 22 Masukkan angka: -1	Jumlah ganjil = 1 Jumlah genap = 4

12. **(P.2 2021)** Buatlah program yang membaca sekumpulan bilangan bulat. Cek apakah angka tersebut lebih besar dari angka sebelumnya. Program akan berhenti menerima data jika angka tidak lebih besar dari angka sebelumnya 3x berturut-turut. Program lalu menampilkan jumlah nilai yang membesar.
Contoh :

Input	Output
Masukkan angka: 2 Masukkan angka: 3 Masukkan angka: 3 Masukkan angka: 2 Masukkan angka: 2	Jumlah nilai yang membesar adalah 3

13. Buatlah program untuk membuat pola siku-siku dengan tinggi N. seperti di bawah (Misal N=5):

```
*
**
***
****
*****
```

14. Buatlah program untuk membuat pola siku-siku dengan tinggi N seperti di bawah (Misal N=3) :

```
  #
 ##
###
```

15. Buatlah program untuk membuat pola segitiga. (Misal N=3):

```
  *
 * *
* * *
```

16. Buatlah program untuk membuat pola rantai sebanyak N seperti di bawah (Misal N=3):

```
* * * * * * *
*   *   *   *
* * * * * * *
```

17. **(P.2 2021)** Buatlah program untuk membuat pola rantai dengan panjang P sebanyak N. (Misal P = 2 dan N = 3) :

```
 ** ** **
*  *  *  *
*  *  *  *
 ** ** **
```

18. **(P.2 2021)** Buatlah program yang menerima 2 angka x dan y. Fungsi f(x,y) adalah 2x+y. Jika f(x,y)<2000, maka nilai x dan y ditambah dengan f(x,y). Tentukan berapa kali operasi penambahan tersebut dilakukan.
Contoh :

Input	Output
Masukkan nilai x: 3 Masukkan nilai y: 4	Dilakukan 5 kali operasi.

Dipindai dengan CamScanner

PEMBAHASAN

1. Untuk menuliskan bilangan 1-n, kita gunakan pengulangan for :

```
1. # Input nilai n :
2. n = int(input("Masukkan n : "))
3. for i in range(1,n+1):
4.     print(f"{i} ")
```

Ingat bahwasanya parameter range menghitung dari a hingga n-1. Sehingga untuk menulis 1-n maka n harus ditambah 1.

2. Maksud dari soal ini adalah kita menentukan bilangan 10^x berapa (misal 10, 100, 1000, dst) yang lebih dari N. Misalkan :

N = 5	10
N = 1234	10000
N = 12345	100000

Untuk itu kita gunakan pengulangan while. Kita cek selama N lebih besar dari 10^x, maka kita tambah x dengan 1.

```
1. # Buat input
2. N = int(input("Masukkan N : "))
3. # Inisiasi x=1 untuk 10^x
4. x = 1
5. # Selama nilai N > 10^x
6. while (N > 10**x):
7.             # Maka x akan ditambah 1
8.             x = x+1
9. # Jika N <= 10^x, maka kita tulis nilai 10^x
10. print(10**x)
```

3. Untuk melakukannya, kita gunakan pengulangan for :

```
1. N = int(input("Masukkan N: "))
2. # Variabel count digunakan untuk menghitung hasil penjumlahan
3. count = 0
4. # Lakukan pengulangan untuk i = 1 hingga N
5. for i in range(1,N+1):
6.             # Filter hanya untuk i bernilai genap
7.             if(i%2==0):
8.                         # Tambah ke count jika nilai i genap
9.                         count+=i
10. print(count)
```

4. Kita akan melakukan iterasi dari 1 hingga N untuk mengecek ada berapa bilangan yang dapat membagi bilangan N. Jika hanya ada 2, berarti ia bilangan prima. Jika tidak, maka bukan bilangan prima.

```
1. N = int(input("Masukkan N: "))
2. # Variabel pembagi digunakan untuk menghitung banyaknya bilangan
     yang dapat membagi N
3. pembagi = 0
4. for i in range(1,N+1):
5.             # Cek apakah i dapat membagi N
6.             if(N%i==0):
```

```
                    # Jika bisa, maka pembagi bertambah
                    pembagi+=1

if(pembagi==2):
                print("Prima")

else:
                print("Bukan prima")
```

5. Gunakan pengulangan for

```
N = int(input("Masukkan N : "))
sum = 0
for i in range(1,N+1):
 if(i%5==0):
      sum+=i
print(sum)
```

6. Implementasikan fungsi tersebut dalam for :

```
# Masukkan seluruh variabel yang dibutuhkan
# Bilangan real = float
# Bilangan bulat = int
a = float(input("Masukkan a: "))
b = float(input("Masukkan b: "))
x = float(input("Masukkan x: "))
n = int(input("Masukkan n : "))
# Variabel sum digunakan untuk menghitung jumlah sigma
sum = 0
for i in range(1,n+1): # Lakukan pengulangan untuk i=1 hingga n
          sum += a*(x**i)+b # Implementasikan fungsi sigma
print(sum)
```

7. Kita implementasikan dengan pengulangan for dari 1 hingga suku ke n

```
n = int(input("Masukkan nilai n: "))
e = 1  # Inisialisasi bilangan Euler awal
faktorial = 1  # Inisialisasi faktorial awal
for i in range(1, n + 1): # Lakukan iterasi dari i=1 hingga deret ke n
    faktorial *= i  # Menghitung faktorial
    e += 1 / faktorial  # Menghitung deret Euler
print(e)
```

Nilai faktorial N! adalah 1x2x3...xN. Kita tidak perlu menghitung nilai faktorialnya satu persatu. Dalam loop for di atas, kita menghitung faktorial dari setiap iterasi dengan mengalikannya dengan nilai i.

8. Implementasikan hasil dari No. 7 dan rumus yang telah diberikan :

```
# Cari nilai e
n = 100
e = 1  # Inisialisasi bilangan Euler awal
faktorial = 1  # Inisialisasi faktorial awal
for i in range(1, n + 1): # Lakukan iterasi dari i=1 hingga deret ke n
    faktorial *= i  # Menghitung faktorial
    e += 1 / faktorial  # Menghitung deret Euler
```


Dipindai dengan CamScanner

```
# Masukkan nilai m :
m = int(input("Masukkan m : "))
for i in range(m):
    Pt = 1.2 * (e**(0.015*i))
    print(f"Tahun ke-{i+1}: {Pt} juta")
```

9. Faktor dari N adalah seluruh angka yang dapat membagi N. Misal 72 faktornya adalah 72, 36, 24, 18, 12, 9, 8, 6, 4, 3, 2, dan 1. Untuk melakukannya, kita tinggal melakukan hitung mundur dari i=N hingga 1, apakah i dapat membagi N. Jika iya, maka tulis nilai i.

```
N = int(input("Masukkan N: "))
for i in range(N,1,-1):
 if (N%i==0):
     print(i)
```

Perhatikan bahwa nilai b pada range(a,n,b) adalah -1, yang artinya setiap iterasi (i) dikurang 1 (hitung mundur).

10. Gunakan for loop dari i=1 hingga N :

```
N = int(input("Masukkan jumlah mahasiswa : "))
# Inisiasi variabel banyak mahasiswa lulus dan tidak lulus
lulus = 0
tdklulus = 0
for i in range(1,N+1):
    Nilai = input() # Lakukan input dari 1 hingga N
            #Jika nilainya A,B,C, atau D, maka lulus ditambah dengan 1
    if (Nilai=="A" or Nilai=="B"or Nilai=="C"or Nilai=="D"):
        lulus += 1
    else: # Jika tidak maka tdklulus ditambah dengan 1
        tdklulus += 1
print(f"Lulus : {lulus}")
print(f"Tidak lulus : {tdklulus}")
```

11. Soal ini bertipe infinite loop, sehingga kita menggunakan while loop. Looping kita hentikan dengan break.

```
# Inisiasi variabel jumlah ganjil dan genap
ganjil = 0
genap = 0
# Lakukan infinite loop dengan while(true)
while(True):
    angka = int(input("Masukkan angka: "))
    if(angka<0):
        break # Jika yang dimasukkan negatif, maka loop dihentikan
    if(angka%2==0): # Cek apakah genap
        genap +=1
    else : # Jika bukan genap, maka pasti ganjil
        ganjil += 1
print(f"Jumlah ganjil = {ganjil}")
print(f"Jumlah genap = {genap}")
```

12. Kita gunakan while loop, selama angka tidak turun selama 3 kali, maka kita akan terus menerima input untuk menjumlahkan angka yang naik.

Dipindai dengan CamScanner

```
before = 0 # Variabel untuk menyimpan nilai sebelumnya
batas = 0 # Hitung berapa kali nilai n tidak naik berturut-turut
sum = 0 # Untuk menghitung jumlah n yang naik dari sebelumnya
i = 1
while(batas!=3): # Selama batasnya tidak sama dengan 3, maka :
    n = int(input("Masukkan angka : ")) # Input untuk  nilai n
    if(n>before and i!=1): # Jika n lebih dan n bukan input pertama
        sum += n # Jika n lebih dari sebelumnya, n dimasukkan ke sum
        batas = 0 # Batas direset menjadi 0
    else:
        batas += 1 # Karena nilai n tidak naik, batas ditambah 1
    before = n #Simpan nilai n untuk dibandingkan dengan berikutnya
    i +=1
print(f"Jumlah nilai yang membesar adalah {sum}.")
```

13. Kita akan menggunakan pengulangan bersarang :

```
1. n = int(input("Masukkan N : "))
2. for i in range(1, n+1): # Iterasi untuk baris i dari 1 hingga n
3.     for j in range(i): # Buat bintang sebanyak i
4.         print("*", end=" ")
5.     print()
```

Pada line 4, parameter (end=" ") berfungsi agar print selanjutnya tetap berada di baris yang sama dan menambah spasi di setiap bintangnya. Pada dasarnya jika kita print sesuatu, maka print selanjutnya akan membuat baris baru. Paramtere end tersebut akan membuat print tidak membuat baris baru. Selanjutnya untuk setiap baris, kita lakukan print() untuk membuat baris baru (dalam mengetik, kita menekan enter).

14. Pada pola ini kita harus menuliskan spasi agar pagar berada di kolom kanan. Misalkan jika N = 5, maka baris pertama kita perlu 4 spasi dan baru 1 pagar. Baris kedua kita perlu 3 spasi dan 1 pagar.

```
#   → (spasi) (spasi) (spasi) (spasi) (pagar)
# #  → (spasi) (spasi) (spasi) (pagar) (pagar)
```

Secara umum, untuk baris ke i, kita perlu n-i spasi dan i pagar. Sehingga dalam iterasi untuk baris i, kita lakukan dua kali iterasi untuk kolom, yaitu untuk spasi dan untuk pagar.

```
n = int(input("Masukkan N :"))
for i in range(1,n+1):
    for j in range(n-i): # Buat spasi
        print(" ", end=" ")
    for k in range(i): # Buat pagar
        print("#", end=" ")
    print() # Buat baris selanjutnya
```

15. Kita analisis pola segitiga tersebut. Misalkan n = 3, s adalah spasi dan b adalah bintang :

```
*          → s s b
* * *      → s b b b
* * * * *  → b b b b b
```


Dipindai dengan CamScanner

Bila kita perhatikan, baris pertama butuh 2 spasi dan 1 bintang. Baris kedua butuh 1 spasi dan 3 bintang. Secara umum diperoleh bahwa untuk baris ke-i, kita butuh (n-i) spasi dan 2i-1 bintang.

```
1. n = int(input("Masukkan N: "))
2. for i in range(1,n+1):
3.     for j in range(n-i):
4.         print(" ", end=" ")
5.     for i in range((2*i)-1):
6.         print("*", end=" ")
7.     print() # Buat baris selanjutnya
```

16. Mari kita analisis pola rantai tersebut.

Misalkan N = 3 :

```
* * * * * * * → (7 bintang)
*   *   *   * → (4 bintang-spasi)
* * * * * * * → (7 bintang)
```

Misalkan N = 2

```
* * * * * → (5 bintang)
*   *   * → (3 bintang-spasi)
* * * * * → (5 bintang)
```

Bila kita perhatikan, pada baris pertama dan ketiga program akan menulis bintang sebanyak (2N+1), dan pada baris kedua sebanyak (N+1) dengan tambahan spasi setelahnya. Implementasikan hasil analisis ke dalam program.

```
1. # Input jumlah rantai
2. N = int(input("Masukkan jumlah rantai: "))
3. # Mencetak pola rantai atas (2n+1 bintang)
4. for j in range((2*N)+1):
5.     print("*", end=" ")
6. print() # Garis baru
7. # Mencetak pola rantai tengah (n+1 bintang-spasi)
8. for j in range(N+1):
9.     print("*", end=" ")
10.    print(" ", end=" ")
11. print() # Garis baru
12. # Mencetak pola rantai bawah (2n+1 bintang)
13. for j in range((2*N)+1):
14.         print("*", end=" ")
15. print() # Garis baru
```

17. Mari kita analisis pola rantai tersebut :

Misalkan panjang (p) = 2 dan banyak rantai (l) = 3 :

```
 * *   * *   * *        → s b b s b b s b b s
*     *     *     *     → b s s b s s b s s b
*     *     *     *     → b s s b s s b s s b
 * *   * *   * *        →
```

Misalkan panjang (p) = 3 dan banyak rantai (l) = 2 :

```
 * * *   * * *          → s b b b s b b b s
*       *       *       → b s s s b s s s b
*       *       *       → b s s s b s s s b
*       *       *       → b s s s b s s s b
 * * *   * * *          → s b b b s b b b s
```

Baris pertama akan membuat s b b b sebanyak banyaknya rantai, dengan banyak b sebanyak panjang rantai, lalu ditutup dengan s. P baris berikutnya akan membuat b s s s sebanyak banyaknya rantai dengan s sebanyak panjang rantai, lalu ditutup dengan b. Baris terakhir akan membuat sama dengan baris pertama.

```
1. p = int(input("Masukkan panjang sisi: "))
2. l = int(input("Masukkan banyak rantai: "))
3. # Baris pertama
4. for i in range(l): # membuat s b b sebanyak l
5.     print(" ",end=" ")
6.     for j in range(p): # dengan b sebanyak p
7.         print("*",end=" ")
8. print(" ", end=" ") # Ditutup dengan s
9. print() # Baris baru
10.
11. # P baris berikutnya
12. for i in range(p):
13.     for j in range(l): # membuat b s s sebanyak l
14.         print("*",end=" ")
15.         for k in range(p): # dengan s sebanyak p
16.             print(" ",end=" ")
17.     print("*",end=" ") # Ditutup dengan b
18.     print() # Baris baru
19. # Baris terakhir, sama dengan baris pertama
20. for i in range(l): # membuat s b b sebanyak l
21.     print(" ",end=" ")
22.     for j in range(p): # dengan b sebanyak p
23.         print("*",end=" ")
24. print(" ", end=" ") # Ditutup dengan s
25. print() # Baris baru
```

18. Berikut pembahasan untuk No. 18 :

```
1. # Input nilai yang dibutuhkan
2. x = int(input("Masukkan nilai x: "))
3. y = int(input("Masukkan nilai y: "))
4. fxy = 0 # Inisiasi fxy dengan awalan 0, untuk menghitung nilai f(x,y)
5. itung = 0 # Untuk menghitung berapa banyak operasi yang dilakukan
6. while(fxy<=2000): # Selama f(x,y) kurang dari 2000
7.     fxy = x + 2*y # Melakukan operasi sesuai fungsi yang diberikan
8.     x = x+fxy # Melakukan operasi sesuai fungsi yang diberikan
9.     y = y+fxy # Melakukan operasi sesuai fungsi yang diberikan
10.    itung = itung + 1 #Menghitung operasi
11. print(f"Dilakukan {itung} kali operasi.") #Mengeluarkan berapa banyak
    operasi yang dilakukan (itung)
```


Dipindai dengan CamScanner

BAB VI : Array dan Matriks

ARRAY

Array dapat diibaratkan dengan himpunan nilai. Misalkan :

```
1. A = [12,95,35,28]
```

Maka A[0] bernilai 12, A[1] bernilai 95, A[2] bernilai 35, dan A[3] bernilai 28. Perlu diperhatikan bahwa indeks array dimulai dari nol.

DEKLARASI ARRAY

- Penulisan langsung :

```
1. A = [12,95,35,28]
```

- Penulisan berulang sebanyak N :

```
1. B = [0 for i in range(N)]
2. C = ["x" for i in range(N)]
```

Nilai B: [0,0,0,...] 0 sebanyak N
Nilai C: ["x", "x",...] "x" sebanyak N

OPERASI ARRAY

Misalkan A = [12,95,35,28]

- Akses array

```
1. x = A[0]
2. y = A[1:3]
```

Variabel x bernilai 12, dan variabel y bernilai [95,35]. Operasi A[a,b] akan menghasilkan array indeks a sampai sebelum b.

- Modifikasi array

```
1. A[2] = 20
```

Array A akan menjadi [12,95,20,28].

- Panjang Array

```
1. x = len(A)
```

Variabel x bernilai 4.

- Menambah nilai pada array

```
1. A.append(59)
```

Array A akan menjadi [12,95,35,28,59]

- Menghapus nilai dengan indeks

```
1. del A[1]
```

Array A akan menjadi [12,35,28]

INPUT ARRAY

Misalkan kita ingin memasukkan data sepuluh bilangan bulat ke array D :

```
1. D = [0 for i in range(10)]
2. for i in range(len(D)):
3.     D[i] = int(input(f"Masukkan
       nilai ke-{i}: "))
```

MATRIKS

Matriks adalah array 2 dimensi, atau array di dalam array. Misalkan :

```
1. E = [[1,2],[3,4]]
```

Nilai E akan menghasilkan matriks :

```
1 2
3 4
```

DEKLARASI MATRIKS

Deklarasi dengan penulisan langsung matriks dapat dilakukan dengan cara seperti di atas. Untuk deklarasi matriks 0 dengan B baris dan K kolom:

```
1.  E = [[0 for i in range(K)] for
    j in range(B)]
```

INPUT MATRIKS

Misalkan matriks E sudah dideklarasikan sebagai matriks bilangan bulat dengan B baris dan K kolom. Maka untuk memasukkan nilai ke matriks E adalah :

```
1. for i in range(B):
2.     for j in range(K):
3.         E[i][j] =
int(input(f"Masukkan elemen E baris
{i+1} kolom {j+1} : "))
```

OUTPUT MATRIKS

Untuk melakukan print matriks E dengan B baris dan K kolom adalah :

```
1. for i in range(B):
2.   for j in range(K):
3.     print(E[i][j], end=" ")
4.   print()
```


Dipindai dengan CamScanner

SOAL-SOAL

1. Buatlah program untuk menerima nilai N, lalu buatlah array A untuk bilangan bulat dengan panjang N dan lakukan penginputan. Contoh :

Input	Output
Masukkan nilai N: 3 Masukkan nilai untuk elemen ke-1: 2 Masukkan nilai untuk elemen ke-2: 4 Masukkan nilai untuk elemen ke-3: 8	[2, 4, 8]

2. (P.3 2021) Ulangi soal No.1 lalu naikkan nilai array sebesar 20%. Keluarkan hasilnya. Contoh :

Input	Output
Masukkan nilai N: 3 Masukkan nilai untuk elemen ke-1: 2 Masukkan nilai untuk elemen ke-2: 4 Masukkan nilai untuk elemen ke-3: 8	[2.4, 4.8, 9.6]

3. Ulangi soal No.1, lalu hitung rata-rata, nilai maksimum dan minimum array tersebut. Contoh :

Input	Output
Masukkan nilai N: 3 Masukkan nilai untuk elemen ke-1: 2 Masukkan nilai untuk elemen ke-2: 4 Masukkan nilai untuk elemen ke-3: 8	Rata-rata : 4.6666667 Maksimum : 8 Minimum : 2

4. (H.3 2021) Ulangi soal No.1, lalu keluarkan nilai setiap array dengan urutan terbalik. Contoh :

Input	Output
Masukkan nilai N: 3 Masukkan nilai untuk elemen ke-1: 2 Masukkan nilai untuk elemen ke-2: 4 Masukkan nilai untuk elemen ke-3: 8	[8, 4, 2]

5. (UTS 2018) Ulangi soal No. 1. Hitunglah banyaknya bilangan bulat positif dan genap dari array A. Contoh :

Input	Output
Masukkan nilai N: 3 Masukkan nilai untuk elemen ke-1: 1 Masukkan nilai untuk elemen ke-2: 4 Masukkan nilai untuk elemen ke-3: -2	Jumlah bilangan positif dan genap : 1

6. Dua buah vektor (x,y,z) diinput dari user. Lakukan penjumlahan dari vektor tersebut dan keluarkan hasilnya. Contoh :

Input	Output
Masukkan nilai X1: 1 Masukkan nilai Y1: 2 Masukkan nilai Z1: 3 Masukkan nilai X2: 4 Masukkan nilai Y2: 5 Masukkan nilai Z2: 6	[5.0, 7.0, 9.0]

Dipindai dengan CamScanner

7. **(H.3 2021)** Buat program yang menerima sebuah string, dan tentukan apakah string tersebut palindrom. *Hint : String adalah array dari karakter (huruf/angka).*

Contoh :

Input	Output
Masukkan kata: kasur	Kata kasur bukan palindrom
Masukkan kata: kasurusak	Kata kasurusak adalah palindrom

8. **(H.3 2021)** Buatlah program yang menerima array A dan B dengan tipe data bilangan bulat, lalu periksa apakah array B merupakan anagram dari array A. Array B dikatakan anagram bila elemen-elemen pada array A bertukar posisi dapat menghasilkan array B. Asumsumsikan nilai elemen berada pada rentang 1 sampai 10.

Contoh :

Input	Output
Masukkan banyaknya elemen A: 3 Masukkan elemen A ke-1: 1 Masukkan elemen A ke-2: 3 Masukkan elemen A ke-3: 2 Masukkan banyaknya elemen B: 3 Masukkan elemen B ke-1: 2 Masukkan elemen B ke-2: 1 Masukkan elemen B ke-3: 3	B adalah anagram dari A

9. **(P.3 2021)** Ulangi soal No.1, asumsikan data yang diinput bernilai 0 dan 1. Hitung berapa banyak pertukaran minimum yang diperlukan untuk membuat semua angka 0 bersebelahan, begitu pula semua angka 1.

Contoh :

Input	Output
Masukkan nilai N : 6 Masukkan bilangan ke-1: 1 Masukkan bilangan ke-2: 0 Masukkan bilangan ke-3: 1 Masukkan bilangan ke-4: 1 Masukkan bilangan ke-5: 0 Masukkan bilangan ke-6: 1	Diperlukan 1 kali penukaran

10. Ulangi soal No.1, dan terima bilangan bulat X. Tuliskan pasangan-pasangan angka yang terdapat pada array tersebut untuk menghasilkan nilai X. Satu pasangan artinya dua buah bilangan yang ada pada array tersebut.

Contoh :

Input	Output
Masukkan nilai N: 6 Masukkan nilai untuk elemen ke-1: 1 Masukkan nilai untuk elemen ke-2: 7 Masukkan nilai untuk elemen ke-3: 2 Masukkan nilai untuk elemen ke-4: 5 Masukkan nilai untuk elemen ke-5: 3 Masukkan nilai untuk elemen ke-6: 8 Masukkan nilai X: 10	(7,3) (2,8) (5,5) (3,7) (8,2)

Dipindai dengan CamScanner

11. (P.3 2021) Seekor bebek bernama Kwek menjalankan usaha laundry. Tiap 1kg cucian akan memakan waktu 1 menit. Kwek memiliki N buah mesin cuci, dan pada hari ini terdapat M cucian, dengan berat berbeda-beda. Bebek mengerjakan cucian sesuai urutan. Setiap mesin cuci yang kosong akan langsung diisi dengan cucian berikutnya. Tentukan waktu yang diperlukan Kwek untuk menyelesaikan semua cuciannya!

Contoh :

Input	Output
Banyak mesin cuci: 2 Banyak cucian: 5 Berat cucian 1: 4 Berat cucian 2: 3 Berat cucian 3: 7 Berat cucian 4: 4 Berat cucian 5: 5	Waktu yang diperlukan adalah 13 menit

12. Buatlah program untuk menerima nilai B dan K sebagai baris dan kolom, lalu buatlah matriks M dari bilangan bulat dengan B baris dan K kolom, dan lakukan penginputan nilai pada matriks tersebut.

Contoh :

Input	Output
Masukkan jumlah baris (B): 3 Masukkan jumlah kolom (K): 3 Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 5 Masukkan elemen baris 1 kolom 3: 3 Masukkan elemen baris 2 kolom 1: 4 Masukkan elemen baris 2 kolom 2: 2 Masukkan elemen baris 2 kolom 3: 6 Masukkan elemen baris 3 kolom 1: 1 Masukkan elemen baris 3 kolom 2: 3 Masukkan elemen baris 3 kolom 3: 4	

13. Ulangi soal No.12, lalu tuliskan matriks tersebut

Contoh :

Input	Output
Masukkan jumlah baris (B): 3 Masukkan jumlah kolom (K): 3 Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 5 Masukkan elemen baris 1 kolom 3: 3 Masukkan elemen baris 2 kolom 1: 4 Masukkan elemen baris 2 kolom 2: 2 Masukkan elemen baris 2 kolom 3: 6 Masukkan elemen baris 3 kolom 1: 1 Masukkan elemen baris 3 kolom 2: 3 Masukkan elemen baris 3 kolom 3: 4	1 5 3 4 2 6 1 3 4

14. Ulangi soal No.12, lalu kurangi nilai setiap elemen dengan 2. Keluarkan matriks tersebut setelah operasi pengurangan.

Dipindai dengan CamScanner

Contoh :

Input	Output
Masukkan jumlah baris (B): 3 Masukkan jumlah kolom (K): 3 Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 5 Masukkan elemen baris 1 kolom 3: 3 Masukkan elemen baris 2 kolom 1: 4 Masukkan elemen baris 2 kolom 2: 2 Masukkan elemen baris 2 kolom 3: 6 Masukkan elemen baris 3 kolom 1: 1 Masukkan elemen baris 3 kolom 2: 3 Masukkan elemen baris 3 kolom 3: 4	2 10 6 8 4 12 2 6 8

15. Ulangi soal No.12, lalu tuliskan matriks dan hasil transpose dari matriks tersebut. Contoh :

Input	Output
Masukkan jumlah baris (B): 3 Masukkan jumlah kolom (K): 3 Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 5 Masukkan elemen baris 1 kolom 3: 3 Masukkan elemen baris 2 kolom 1: 4 Masukkan elemen baris 2 kolom 2: 2 Masukkan elemen baris 2 kolom 3: 6 Masukkan elemen baris 3 kolom 1: 1 Masukkan elemen baris 3 kolom 2: 3 Masukkan elemen baris 3 kolom 3: 4	Matriks : 1 5 3 4 2 6 1 3 4 Matriks transpose : 1 4 1 5 2 3 3 6 4

16. **(H.4 2021)** Ulangi soal No. 12, lalu tuliskan berapa banyak bilangan positif pada matriks tersebut. Contoh :

Input	Output
Masukkan jumlah baris (B): 3 Masukkan jumlah kolom (K): 3 Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 5 Masukkan elemen baris 1 kolom 3: -3 Masukkan elemen baris 2 kolom 1: 4 Masukkan elemen baris 2 kolom 2: -2 Masukkan elemen baris 2 kolom 3: 6 Masukkan elemen baris 3 kolom 1: 1 Masukkan elemen baris 3 kolom 2: 3 Masukkan elemen baris 3 kolom 3: 4	Ada 7 bilangan positif di matriks. 1 5 -3 4 -2 6 1 3 4

17. Buatlah program yang menerima matriks M dengan ukuran 3x3, dan lakukan penginputan nilai. Tentukan determinan dari matriks tersebut. Contoh :

Input	Output
Masukkan jumlah baris (B): 3 Masukkan jumlah kolom (K): 3	Determinan dari matriks tersebut adalah -30

Dipindai dengan CamScanner

Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 5 Masukkan elemen baris 1 kolom 3: 3 Masukkan elemen baris 2 kolom 1: 4 Masukkan elemen baris 2 kolom 2: 2 Masukkan elemen baris 2 kolom 3: 6 Masukkan elemen baris 3 kolom 1: 1 Masukkan elemen baris 3 kolom 2: 3 Masukkan elemen baris 3 kolom 3: 4	

18. (H.4 2021) Buatlah sebuah segitiga pascal dengan matriks NxN. Contoh :

Input	Output
Masukkan N : 3	1 1 1 1 1 2 3 4 1 3 6 10 1 4 10 20

19. Buatlah program yang menerima matriks M1 dengan ukuran B1 baris dan K1 kolom, dan matriks M2 dengan B2 baris dan K2 kolom. Jumlahkan matriks M1 dan M2 dan keluarkan hasilnya.

Contoh :

Input	Output
Masukkan baris matriks M1: 2 Masukkan kolom matriks M1: 2 Masukkan nilai untuk matriks M1: Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 2 Masukkan elemen baris 2 kolom 1: 3 Masukkan elemen baris 2 kolom 2: 4 Masukkan baris matriks M2: 2 Masukkan kolom matriks M2: 2 Masukkan nilai untuk matriks M2: Masukkan elemen baris 1 kolom 1: 3 Masukkan elemen baris 1 kolom 2: 4 Masukkan elemen baris 2 kolom 1: 5 Masukkan elemen baris 2 kolom 2: 6	Hasil penjumlahan M1 + M2: 4 6 8 10

20. Ulangi soal No.11, tentukan apakah matriks tersebut adalah matriks identitas atau bukan

Contoh :

Input	Output
Masukkan jumlah baris (B): 3 Masukkan jumlah kolom (K): 3 Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 0 Masukkan elemen baris 1 kolom 3: 0 Masukkan elemen baris 2 kolom 1: 0 Masukkan elemen baris 2 kolom 2: 1 Masukkan elemen baris 2 kolom 3: 0 Masukkan elemen baris 3 kolom 1: 0 Masukkan elemen baris 3 kolom 2: 0 Masukkan elemen baris 3 kolom 3: 1	Matriks tersebut matriks identitas

Dipindai dengan CamScanner

21. Buatlah program yang menerima matriks A dan matriks B dengan ukuran NxN. Kalikan matriks A dan B dan keluarkan hasilnya.

Contoh :

Input	Output
Masukkan N : 2 Masukkan elemen baris 1 kolom 1: 1 Masukkan elemen baris 1 kolom 2: 2 Masukkan elemen baris 2 kolom 1: 3 Masukkan elemen baris 2 kolom 2: 4 Masukkan nilai untuk matriks B: Masukkan elemen baris 1 kolom 1: 3 Masukkan elemen baris 1 kolom 2: 4 Masukkan elemen baris 2 kolom 1: 5 Masukkan elemen baris 2 kolom 2: 6	Hasil perkalian matriks A dan B: 13 16 29 36

22. **(P.4 2021)** Kwik memiliki sebuah denah dalam bentuk matriks 5x5. Setiap elemen matriks tersebut menunjukkan ketinggian landmark tersebut. Bantulah Kwik untuk menentukan banyak balok yang ia butuhkan dalam membangun landmark.

Contoh :

Input	Output
Masukkan denah: 10001 12221 12321 12221 10001	Balok yang dibutuhkan sebanyak 29 balok.

23. **(P.4 2021)** Kwok ingin membuat papan permainan SOS. Papan itu berbentuk matriks berukuran BxK yang berisi huruf S dan O. Kwok akan menginputkan huruf S dan O pada setiap elemen matriks. Tentukan ada berapa banyak SOS, baik horizontal maupun vertikal pada papan tersebut. Contoh :

Input	Output
Masukkan jumlah baris (B) denah: 3 Masukkan jumlah kolom (K) denah: 3 Masukkan denah (gunakan huruf S dan O): Masukkan elemen baris 1 kolom 1: S Masukkan elemen baris 1 kolom 2: O Masukkan elemen baris 1 kolom 3: O Masukkan elemen baris 2 kolom 1: S Masukkan elemen baris 2 kolom 2: S Masukkan elemen baris 2 kolom 3: O Masukkan elemen baris 3 kolom 1: S Masukkan elemen baris 3 kolom 2: O Masukkan elemen baris 3 kolom 3: S	Jumlah SOS secara horizontal: 1 Jumlah SOS secara vertikal: 0 Total jumlah SOS: 1

24. **(UTS 2018)** Buatlah program yang menerima data berat badan 50 siswa (asumsikan data tidak ada yang salah dan data adalah bilangan bulat). Tampilkan berat badan terbesar. Setelah itu, terima input berat badan minimum. Tampilkan berapa banyak siswa yang melebihi berat badan minimum. Tampilkan juga daftar berat badan yang lebih dari minimum tersebut. Contoh :

Dipindai dengan CamScanner

Input dan Output

```
Masukkan berat badan siswa :
45
64
...
Berat badan tebesar = 83
Masukkan berat badan minimum : 60
Banyak siswa dengan berat > 60 = 14
Daftar berat badan > 60 :
72
83
...
```

Dipindai dengan CamScanner

PEMBAHASAN

1. Soal ini adalah soal dasar untuk membuat dan menginput array.

```
# Inisiasi array
N = int(input("Masukkan nilai N: "))
A = [0 for i in range(N)]
# Pembacaan array
for i in range(N):
    A[i]=int(input(f"Masukkan nilai untuk elemen  ke-{i+1}: "))
# Keluarkan hasil
print(A)
```

2. Lakukan pembacaan array seperti No.1. Untuk menaikkan nilai array sebesar 20%, kita kalikan tiap elemennya dengan 1.2 menggunakan pengulangan.

```
# Inisiasi array
N = int(input("Masukkan nilai N: "))
A = [0 for i in range(N)]
# Pembacaan array
for i in range(N):
    A[i]=int(input(f"Masukkan nilai untuk elemen  ke-{i+1}: "))
# Kalikan tiap elemen dengan 1.2
for i in range(N):
    A[i] = A[i]*1.2
print(A)
```

3. Untuk mencari nilai maksimum dan minimum dari array, kita dapat menggunakan fungsi max dan min. Misalkan A adalah sebuah array, maka max(A) adalah nilai maksimum dari array, dan min(A) adalah nilai minimum dari array. Sedangkan untuk mencari rata-rata, kita gunakan fungsi sum untuk mendapatkan hasil penjumlahan elemen-elemen array tersebut, dan len untuk mendapatkan banyak elemen pada array tersebut. Sehingga, misalkan A adalah sebuah array, rata-rata dapat dicari dengan sum(A)/len(A).

```
# Inisiasi array
N = int(input("Masukkan nilai N: "))
A = [0 for i in range(N)]
# Pembacaan array
for i in range(N):
    A[i]=int(input(f"Masukkan nilai untuk elemen  ke-{i+1}: "))

# Cari rata-rata
ratarata = sum(A)/len(A)
# Cari nilai maksimum
maksimum = max(A)
# Cari nilai minimum
minimum = min(A)
```


Dipindai dengan CamScanner

```
14. #Keluarkan hasil
15. print(f"Rata-rata : {ratarata}")
16. print(f"Maksimum : {maksimum}")
17. print(f"Minimum : {minimum}")
```

4. Untuk membuat array terbalik, kita buat array lain dengan jumlah elemen yang sama. Setelah itu kita lakukan iterasi sehingga :
A_balik[i] = A[(N-1)-i]
Perhatikan contoh berikut :
N = 3

Indeks (i)	0	1	2
A[i]	5	10	15
((N-1)-i)	(3-1)-0 = 2	(3-1)-1 = 1	(3-1)-2 = 0
A_balik[i] = A[(N-1)-i]	15	10	5

Implementasi dalam program python-nya adalah sebagai berikut :

```
 1. # Inisiasi array
 2. N = int(input("Masukkan nilai N: "))
 3. A = [0 for i in range(N)]
 4. # Pembacaan array
 5. for i in range(N):
 6.     A[i]=int(input(f"Masukkan nilai untuk elemen  ke-{i+1}: "))
 7. A_balik = [0 for i in range(N)]
 8. # Buat array secara terbalik :
 9. for i in range(N):
10.     A_balik[i] = A[(N-1)-i]
11. print(A_balik)
```

5. Untuk mengecek apakah suatu bilangan genap, gunakan modulo 2. Jika hasilnya 0, bilangan tersebut genap. Jika hasilnya 1, maka bilangan tersebut ganjil.

```
 1. # Inisiasi array
 2. N = int(input("Masukkan nilai N: "))
 3. A = [0 for i in range(N)]
 4. # Pembacaan array
 5. for i in range(N):
 6.     A[i]=int(input(f"Masukkan nilai untuk elemen  ke-{i+1}: "))
 7. # Hitung nilai positif dan genap
 8. positifgenap = 0
 9. for i in range(N):
10.     if (A[i]%2==0 and A[i]>0): # Jika A[i] genap dan positif
11.         positifgenap += 1
12. # Keluarkan hasil
13. print(f"Jumlah bilangan positif dan genap : {positifgenap}")
```

6. Pertama-tama kita harus menginisiasi array V1 dan V2 dengan jumlah elemen 3. Lalu lakukan input tiap elemen untuk V1 dan V2 tersebut. Setelah itu, baru kita dapat lakukan penjumlahan dengan membuat array V3 = V1 + V2

Dipindai dengan CamScanner

```
# Inisiasi array
V1 = [0.0 for i in range(3)]
V2 = [0.0 for i in range(3)]

# Lakukan input
V1[0] = float(input("Masukkan nilai X1: "))
V1[1] = float(input("Masukkan nilai Y1: "))
V1[2] = float(input("Masukkan nilai Z1: "))
V2[0] = float(input("Masukkan nilai X2: "))
V2[1] = float(input("Masukkan nilai Y2: "))
V2[2] = float(input("Masukkan nilai Z2: "))

# Buat array V3 sebagai V3=V1+V2
V3 = [0.0 for i in range(3)]
V3[0] = V1[0]+V2[0]
V3[1] = V1[1]+V2[1]
V3[2] = V1[2]+V2[2]
print(V3)
```

7. Kata palindrom adalah kata yang apabila huruf-hurufnya dibalik maka hasilnya akan sama. Misalkan kata "APA", jika dibalik akan tetap "APA". Misalkan n adalah jumlah hurufnya. Untuk mengeceknya, kita periksa apakah huruf ke-i sama dengan huruf ke-(n-i). Perlu diingat bahwa array dimulai dari nol, sehingga kita perlu kurangi n dengan 1. Jadi yang kita bandingkan adalah huruf ke-i dengan huruf ke-((n-1)-i).

```
# Pembacaan string dan panjang string
A = input("Masukkan kata: ")
n = len(A)

# Pengecekan palindrom
palindrom = True
for i in range(n//2):
    if A[i] != A[(n-1)-i]:
        palindrom=False

# Keluarkan hasil
if (palindrom):
    print(f"kata {A} adalah palindrom")
else:
    print(f"Kata {A} bukan palindrom")
```

String sebenarnya adalah array dari huruf. Misalkan string A = "APA", maka sebetulnya itu adalah array A = [A,P,A]. Kita menggunakan pembenar dengan bool palindrom dan diinisiasi menjadi True. Jika ada A[i] yang tidak sama dengan A[(n-1)-i], maka pembenar bernilai False.

8. Array A dapat dikatakan anagram terhadap array B apabila elemen-elemen array A diubah susunannya maka akan sama dengan array B. Misal A = [1,2,3] dan B = [2,1,3], maka A adalah anagram dari B. Karena asumsi elemen array hanya dalam

Dipindai dengan CamScanner

rentang 1-10, maka kita gunakan saja tabel frekuensi. Misalkan array C adalah tabel frekuensi. Maka C[i] adalah jumlah angka i+1 pada array A.

Indeks	0	1	2	3
A	4	8	3	4

Indeks	0	1	2	3	4	5	6	7	8	9
C	0	0	1	2	0	0	0	1	0	0

Penjelasan : Nilai C[3] adalah jumlah angka 4 (3+1) pada array A.

```
1. Implementasi alur berpikir di atas dalam program python adalah :
2. # Array A dan pemasukan nilai
3. A = [0 for i in range(10)] # Inisiasi tabel frekuensi A
4. for i in range(int(input("Masukkan banyaknya elemen A: "))): #
   Iterasi penginputan nilai
5.     x = int(input(f"Masukkan elemen A ke-{str(i+1)}: "))
6.     A[x-1] += 1 # Menambahkan banyak bilangan x ke tabel frekuensi A
7.
8. # Array B dan pemasukan nilai
9. B = [0 for i in range(10)] # Inisiasi tabel frekuensi B
10. for i in range(int(input("Masukkan banyaknya elemen B: "))): #
    Iterasi penginputan nilai
11.     x = int(input(f"Masukkan elemen B ke-{str(i+1)}+: "))
12.     B[x-1] += 1 # Menambahkan banyak bilangan x ke tabel frekuensi B
13.
14. # Cek apakah hasilnya sama
15. if (A==B):
16.     print("B adalah anagram dari A")
17. else:
18.     print("B bukan anagram dari A")
```

9. Permasalahan ini memiliki dua kasus. Kasus pertama adalah pengelompokan dengan angka 1 berada di kiri, dan angka 1 berada di kanan. Misalkan A = [1,0,1,0,1,1,1] dapat dikelompokan dengan angka 1 berada di kiri menjadi [1,1,1,1,1,0,0] dengan 2 kali penukaran, juga dapat dikelompokkan dengan angka 1 berada di kanan menjadi [0,0,1,1,1,1,1] dengan 1 kali penukaran. Maka kita cari berapa kali penukaran untuk dua kasus tersebut dan ambil penukaran yang paling sedikit.
Untuk mencari berapa kali penukaran, kita menghitung berapa banyak angka 1 pada array A sebagai n. Setelah itu hitung angka nol pada n elemen pertama pada array A. Untuk kasus angka 1 berada di kanan maka lakukan kebalikannya yaitu menghitung angka nol pada n elemen terakhir pada array A.

```
1. # Penginputan nilai
2. N = int(input("Masukkan nilai N : "))
3. A = [0 for i in range(N)]
4. for i in range(N):
5.     A[i] = int(input(f"Masukkan bilangan ke-{i+1}: "))
6. # Hitung jumlah angka 1 pada array
7. Jumlah1 = A.count(1)
8. # Kasus angka 1 di depan
```


Dipindai dengan CamScanner

```
depan = 0
for i in range(Jumlah1):
    if(A[i]!=1):
        depan+=1
# Kasus angka 1 di belakang
belakang = 0
for i in range(N-1,(N-1)-Jumlah1,-1):
    if(A[i]!=1):
        belakang+=1
# Cari pertukaran yang paling minimum
if(depan>belakang):
    print(f"Diperlukan {belakang} kali pertukaran.")
else:
    print(f"Diperlukan {depan} kali pertukaran.")
```

10. Untuk melakukannya kita menggunakan metode brute-force, yaitu mencoba satu per satu pasangan angka. Misalkan A = [1,2,3], dan x = 4, maka kita coba dari (1+1) hingga (3+3) : (1+1), (1+2), (1+3), (2+1), (2+2), (2+3), (3+3). Didapatkan bahwa yang menghasilkan angka 4 adalah (1+3), (2+2). Gunakan pengulangan bersarang untuk mengecek A[i] + A[j] == x.

```
# Inisiasi array
N = int(input("Masukkan nilai N: "))
A = [0 for i in range(N)]
# Pembacaan array
for i in range(N):
    A[i]=int(input(f"Masukkan nilai untuk elemen  ke-{i+1}: "))
# Pembacaan X
X = int(input("Masukkan nilai X: "))
# Cari pasangan nilai
for i in range(N):
    for j in range(N):
        if(A[i]+A[j]==X):
            print(f"({A[i]},{A[j]})")
```

11. Pertama kita harus menginput nilai banyaknya mesin cuci dan cucian. Lalu kita buat array cucian untuk menampung banyak cuciannya. Setiap elemen i melambangkan berat cucian ke-i. MesinCuci untuk menampung berapa sisa waktu yang diperlukan mesin cuci. Setiap elemen ke-i menentukan sisa waktu yang diperlukan mesin cuci ke-i untuk menyelesaikan tugasnya.
Setiap 1kg cucian akan memakan waktu 1 menit. Maka kita buat variabel stopwatch untuk menghitung menit. Lalu buat loop selama cucian masih ada dan lakukan penambahan menit pada stopwatch setiap loopnya. Di dalam loop itu, kurangi menit untuk setiap mesin cuci. Kita cek juga untuk setiap mesin cuci, jika sisa waktu 0 (tidak ada cucian/cucian selesai), maka kita tambahkan cucian ke dalam mesin cuci. Apabila cucian sudah tidak ada lagi, maka kita pecahkan pengulangannya dengan break.

Dipindai dengan CamScanner

```
# Input jumlah mesin cuci dan jumlah cucian
N = int(input("Banyak mesin cuci: "))
M = int(input("Banyak cucian: "))

# Input berat cucian
cucian = [0 for i in range(M)]
for i in range(M):
    cucian[i] = int(input(f"Berat cucian {i+1}: "))
# Inisiasi mesin cuci dan waktu
MesinCuci = [0 for i in range(N)]
stopwatch=0
# Loop selama cucian masih ada
while(len(cucian)>0):
    for i in range(N): # Untuk setiap mesin cuci
        if(MesinCuci[i]==0): #Jika ada mesin cuci yang kosong
            MesinCuci[i]+=cucian[0] # Tambah cucian urutan pertama
            del cucian[0] # Hapus cucian urutan pertama
            # Urutan kedua akan menjadi urutan pertama
        MesinCuci[i]-=1 # Kurangi menit untuk tiap mesin cuci
        if(len(cucian)==0): # Jika sudah tidak ada cucian, hentikan loop
            break
    stopwatch+=1 # Tambah menit pada stopwatch
# Tambahkan waktu dengan sisa waktu terlama pada mesin cuci
stopwatch += max(MesinCuci)
print(f"Waktu yang diperlukan adalah {stopwatch} menit")
```

12. Soal ini adalah soal dasar untuk membuat dan menginput matriks

```
# Input B dan K
B = int(input("Masukkan jumlah baris (B): "))
K = int(input("Masukkan jumlah kolom (K): "))
# Inisiasi matriks
A = [[0 for i in range(K)] for j in range(B)]
# Penginputan matriks
for i in range(B):
    for j in range(K):
        A[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom {j+1}: "))
```

13. Untuk mengeluarkan matriks caranya adalah sebagai berikut :

```
# Input B dan K
B = int(input("Masukkan jumlah baris (B): "))
K = int(input("Masukkan jumlah kolom (K): "))
# Inisiasi matriks
A = [[0 for i in range(K)] for j in range(B)]
# Penginputan matriks
for i in range(B):
    for j in range(K):
        A[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom
```


Dipindai dengan CamScanner

```
{j+1}: "))
# Tuliskan matriks
for i in range(B):
    for j in range(K):
        print(A[i][j], end=" ")
    print()
```

14. Gunakan loop bersarang untuk mengalikan tiap elemen matriks dengan 2

```
# Input B dan K
B = int(input("Masukkan jumlah baris (B): "))
K = int(input("Masukkan jumlah kolom (K): "))
# Inisiasi matriks
A = [[0 for i in range(K)] for j in range(B)]
# Penginputan matriks
for i in range(B):
    for j in range(K):
        A[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom
{j+1}: "))
# Kali elemen dengan 2
for i in range(B):
    for j in range(K):
        A[i][j] = A[i][j]*2

# Tuliskan matriks
for i in range(B):
    for j in range(K):
        print(A[i][j], end=" ")
    print()
```

15. Transpose matriks adalah mengubah baris menjadi kolom, dan sebaliknya. Jika ada matriks BxK, kita tinggal membuat matriks baru dengan ukuran KxB. Lalu untuk baris i dan kolom j, kita tukar baris dengan kolom sehingga transpose_A[j][i] = A[i][j].

```
# Input B dan K
B = int(input("Masukkan jumlah baris (B): "))
K = int(input("Masukkan jumlah kolom (K): "))
# Inisiasi matriks
A = [[0 for i in range(K)] for j in range(B)]
# Penginputan matriks
for i in range(B):
    for j in range(K):
        A[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom
{j+1}: "))
# Buat matriks transpose
transpose_A = [[0 for i in range(B)] for j in range(K)] # Matriks
hasil transpose
for i in range(B):
    for j in range(K):
```


Dipindai dengan CamScanner

```
14.             transpose_A[j][i] = A[i][j]
15.
16. # Tuliskan matriks
17. print("Matriks : ")
18. for i in range(B):
19.     for j in range(K):
20.         print(A[i][j], end=" ")
21.     print()
22. print("Matriks transpose : ")
23. # Tuliskan matriks transpose
24. for i in range(K):
25.     for j in range(B):
26.         print(transpose_A[i][j], end=" ")
27.     print()
```

16. Ada tiga tahap untuk menjawab soal ini. Pertama, lakukan penginputan matriks NxM seperti No. 12. Lalu, hitung elemen positif dengan melakukan loop untuk tiap elemennya, dan cek untuk bilangan positif. Catat dengan menambahkan 1 pada variabel sum. Ketiga, tulis ulang matriks seperti No. 13.

```
1. # Lakukan penginputan matriks NxM
2. N = int(input("Masukkan N: ")) # Masukan nilai N
3. M = int(input("Masukkan M: ")) # Masukan nilai M
4. A = [[0 for j in range(M)] for i in range(N)] # Inisiasi matriks A MxN
5. for i in range(N):
6.     for j in range(M):
7.         A[i][j] = int(input(f"Masukkan nilai A[{str(i+1)}][{str(j+1)}]: "))
8.
9. # Hitung bilangan positif
10. sum = 0
11. for i in range(N):
12.     for j in range(M):
13.         if (A[i][j]>0): # Jika ada nilai matriks lebih dari 0
14.             sum += 1
15. print(f"Ada {str(sum)} bilangan positif di matriks.")
16.
17. # Tulis ulang matriks
18. for i in range(N):
19.     for j in range(M):
20.         print(A[i][j],end=" ")
21.     print("")
```


Dipindai dengan CamScanner

17. Determinan dari matriks 3x3 adalah sebagai berikut :

$$\begin{vmatrix} M_{0.0} & M_{0.1} & M_{0.2} \\ M_{1.0} & M_{1.1} & M_{1.2} \\ M_{2.0} & M_{2.1} & M_{2.2} \end{vmatrix} \begin{matrix} M_{0.0} & M_{0.1} \\ M_{1.0} & M_{1.1} \\ M_{2.0} & M_{2.1} \end{matrix}$$

[illegible]

```
1. # Menerima input nilai untuk matriks M
2. M = [[0 for i in range(3)] for j in range(3)]
3. for i in range(3):
4.     for j in range(3):
5.         M[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom
   {j+1}: "))
6. # Hitung determinan
7. determinan_M = ((M[0][0]*M[1][1]*M[2][2]) + (M[0][1]*M[1][2]*M[2][0])
   + (M[0][2]*M[1][0]*M[2][1]))
8. determinan_M -= ((M[0][1]*M[1][0]*M[2][2]) + (M[0][0] * M[1][2] *
   M[2][1]) + (M[0][2]*M[1][1]*M[2][0]))
9. print(f"Determinan dari matriks tersebut adalah {determinan_M}")
```

18. Perhatikan segitiga pascal berikut untuk N = 4 :

```
1 1 1 1
1 2 3 4
1 3 6 10
1 4 10 20
```

Jika kita perhatikan, kecuali untuk baris 1 atau kolom 1, tiap elemen adalah hasil penjumlahan atas + kiri. Misalnya baris 3 kolom 3, nilainya adalah 6. Hal itu berasal dari penjumlahan elemen baris 2 kolom 3 yang bernilai 3, dan baris 3 kolom 2 yang bernilai 3, sehingga 3+3 = 6.

```
1. # Input nilai N
2. N = int(input("Masukkan N: "))
3.
4. # Inisiasi matriks NxN dengan nilai 1 semua
5. SP = [[1 for j in range(N)] for i in range(N)]
6.
7. # Buat segitiga pascal
8. for i in range(N):
9.     for j in range(N):
10.         # Pengecualian untuk baris pertama atau kolom pertama
11.         if (i!=0 and j!=0):
12.
13. # Hitung menggunakan rumus (nilai = atas + kiri)
14.             SP[i][j] = SP[i-1][j] + SP[i][j-1]
15.
```


Dipindai dengan CamScanner

```
# Tulis matriks
for i in range(N):
    for j in range(N):
        print(SP[i][j], end=" ")
    print("")
```

19. Pertama kita harus input matriks M1 dan M2. Setelah itu pastikan dimensi matriks M1 dan M2 sama. Barulah kita jumlahkan setiap elemen dari kedua matriks tersebut dengan membuat matriks baru (M_{sum}) dan keluarkan matriks tersbeut.

```
# Input matriks M1
B1 = int(input("Masukkan baris matriks M1: "))
K1 = int(input("Masukkan kolom matriks M1: "))
print("Masukkan nilai untuk matriks M1:")
M1 = [[0 for i in range(K1)] for j in range(B1)]
for i in range(B1):
  for j in range(K1):
      M1[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom {j+1}:
"))

# Input matriks M2
B2 = int(input("Masukkan baris matriks M2: "))
K2 = int(input("Masukkan kolom matriks M2: "))
print("Masukkan nilai untuk matriks M2:")
M2 = [[0 for i in range(K2)] for j in range(B2)]
for i in range(B2):
  for j in range(K2):
      M2[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom {j+1}:
"))

# Menyusun matriks hasil penjumlahan
if B1 == B2 and K1 == K2: # Pastikan ukuran matriks sama
  M_sum = [[0 for i in range(K1)] for j in range(B1)]
  # Jumlahkan tiap elemen
  for i in range(B1):
      for j in range(K1):
          M_sum[i][j] = M1[i][j] + M2[i][j]
  # Keluarkan hasil
  print("Hasil penjumlahan M1 + M2:")
  for i in range(B1):
      for j in range(K1):
          print(M_sum[i][j], end=" ")
      print()
else:
  print("Ukuran matriks M1 dan M2 tidak cocok untuk penjumlahan.")
```

20. Matriks dapat dikatakan identitas apabila diagonal utamanya bernilai 1 dan elemen lainnya bernilai 0. Gunakan pembenar dengan bool identitas, lalu iterasi apakah diagonal utamanya bernilai 1 dan elemen lainnya bernilai 0. Jika ada yang

Dipindai dengan CamScanner

salah, maka bool identitas diubah menjadi False. Elemen baris i kolom j adalah elemen diagonal jika i = j.

```
# Input B dan K
B = int(input("Masukkan jumlah baris (B): "))
K = int(input("Masukkan jumlah kolom (K): "))
# Inisiasi matriks
A = [[0 for i in range(K)] for j in range(B)]
# Penginputan matriks
for i in range(B):
    for j in range(K):
        A[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom {j+1}: "))
# Cek matriks identitas
identitas = True
for i in range(B):
    for j in range(K):
        if i == j and A[i][j] != 1:  # Elemen di diagonal utama harus bernilai 1
            identitas = False
        elif i != j and A[i][j] != 0:  # Elemen di luar diagonal utama harus bernilai 0
            identitas = False
# Keluarkan hasil
if identitas:
    print("Matriks tersebut matriks identitas")
else:
    print("Matriks tersebut bukan matriks identitas")
```

21. Untuk mengalikan matriks dengan matriks, maka caranya adalah sebagai berikut :

Untuk 2x2 : ẅ ẃ z Ẅ Ẅ ẅẄ ẃẄẅẄẅẄẃẄ
ẃ Ẅ Ẅ Ẅ ẃẄẄẄẃẄẄẄ

Untuk 3x3 : ẅ ẃ ẃ ṅ ṅ ṅ ẅṅ ẃṅ ẃṅ ẅṅ ẃṅ ẃṅ ẅṅ ẃṅ ẃṅ
Ẅ Ẅ Ẅ z ṅ ṅ ṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ
Ẅ Ẅ Ẅ ṅ ṅ ṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ Ẅṅ

Misalkan matriks C adalah hasil kali matriks A dan B dengan ukuran NxN. Bila kita cermati untuk setiap baris ke-i dan kolom ke-j, akan ada penjumlahan sebanyak N kali. Maka kita buat iterasi k untuk setiap penjumlahan. Sehingga C[i][j] = A[i][k]*B[k][j].

```
# Input matriks A
N = int(input("Masukkan N : "))
A = [[0 for i in range(N)] for j in range(N)]
for i in range(N):
    for j in range(N):
        A[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom {j+1}: "))
```

```
8. # Input matriks B
9. print("Masukkan nilai untuk matriks B:")
10. B = [[0 for i in range(N)] for j in range(N)]
11. for i in range(N):
12.   for j in range(N):
13.     B[i][j] = int(input(f"Masukkan elemen baris {i+1} kolom {j+1}:
    "))
14.
15. # Lakukan perkalian matriks
16. C = [[0 for i in range(N)] for j in range(N)]
17. for i in range(N):
18.   for j in range(N):
19.       for k in range(N):
20.           C[i][j] += A[i][k] * B[k][j]
21. # Keluarkan hasilnya
22. print("Hasil perkalian matriks A dan B:")
23. for i in range(N):
24.   for j in range(N):
25.     print(C[i][j], end=" ")
26.   print()
```

22. Kita hanya perlu menjumlahkan semua elemen matriks tersebut. Kita gunakan string untuk input setiap barisnya. Ingat bahwa string adalah array of char. Setelah itu, kita konversi tiap angka pada string tersebut ke int, lalu jumlahkan hasilnya.

```
1. denah = ["" for i in range(5)] # Inisiasi array of string panjang 5
2. print("Masukkan denah:")
3. for i in range(5): # Input denah tiap baris sebagai array of string
4.     denah[i]=input()
5. # Menghitung jumlah balok yang dibutuhkan setiap petak
6. total = 0
7. for i in range(5):
8.     for j in range(5):
9.         total += int(denah[i][j])
10. print(f"Balok yang dibutuhkan sebanyak {total} balok.")
```

23. Permainan ini akan menghitung SOS dalam dua kasus yaitu secara horizontal dan vertikal. Sehingga kita bagi kasus menjadi dua.

```
1. # Lakukan  penginputan
2. B = int(input("Masukkan jumlah baris (B) denah: "))
3. K = int(input("Masukkan jumlah kolom (K) denah: "))
4. denah = [["" for i in range(K)] for i in range(B)]
5. print("Masukkan denah (gunakan huruf S dan O):")
6. for i in range(B):
7.   for j in range(K):
8.     denah[i][j] = input(f"Masukkan elemen baris {i+1} kolom {j+1}: ")
9.
```


Dipindai dengan CamScanner

```
# Hitung jumlah SOS secara horizontal
total_sos_horizontal = 0
for i in range(B):
    for j in range(K-2):
        if denah[i][j]=='S' and denah[i][j+1]=='O' and denah[i][j+2]=='S':
            total_sos_horizontal += 1

# Hitung jumlah SOS secara vertikal
total_sos_vertikal = 0
for i in range(B-2):
    for j in range(K):
        if denah[i][j]=='S' and denah[i+1][j]=='O' and denah[i+2][j]=='S':
            total_sos_vertikal += 1

# Total jumlah SOS
total_sos = total_sos_horizontal + total_sos_vertikal

# Keluarkan hasil
print(f"Jumlah SOS secara horizontal: {total_sos_horizontal}")
print(f"Jumlah SOS secara vertikal: {total_sos_vertikal}")
print(f"Total jumlah SOS: {total_sos}")
```

24. Ada beberapa tahap yang perlu dilalui. Pertama, lakukan inisiasi dan penginputan array A seperti No. 1, sebagai himpunan dari berat badan 50 siswa. Kedua, cari nilai terbesar dari array seperti No. 3. Ketiga, masukkan nilai minimum. Keempat, cari nilai pada array A yang lebih dari nilai minimum yang dimasukkan. Caranya adalah dengan membuat array kosong B, lalu gunakan fungsi B.append(X) untuk memasukkan nilai X pada array B. Kita menggunakan fungsi append karena kita tidak memiliki kepastian informasi mengenai panjang array. Kelima, tulis banyak siswa dengan berat yang lebih dari minimum dengan mencari panjang array B, serta tuliskan daftar array B.

```
# Inisiasi dan penginputan array
n = 20
A = [0 for i in range(n)]
print("Masukkan berat badan siswa : ")
for i in range(n):
    A[i] = int(input())

# Cari berat badan terbesar
terbesar = max(A)
print(f"Berat badan terbesar: {terbesar}")

# Input nilai minimum
minimum = int(input("Masukkan berat badan minimum : "))

```


Dipindai dengan CamScanner

```
15. # Cari dan masukan dalam array berat yang lebih besar dari minimum
16. B = [] # Array untuk menampung nilai
17. for i in range(n):
18.     if (A[i]>minimum):
19.         B.append(A[i])
20.
21. # Tulis banyak siswa dengan berat > minimum
22. print(f"Banyak siswa dengan berat > {minimum} = {len(B)}")
23.
24. # Tulis berat badan seluruh siswa dengan berat > minimum
25. print(f"Daftar berat badan > {minimum} : ")
26. for i in range(len(B)):
27.     print(B[i])
```


Dipindai dengan CamScanner

BAB VII: Fungsi dan Prosedur

FUNGSI

Fungsi adalah sebagian program yang terpisah dari program utama untuk melakukan tugas tertentu dengan menerima parameter dan mengembalikan hasil. Contoh :

```
1. def luas_segitiga(alas, tinggi):
2.       Luas = 0.5 * alas * tinggi
3.       return Luas
```

Fungsi luas_segitiga di atas adalah fungsi untuk menghitung luas segitiga. Parameternya adalah alas dan tinggi. Fungsi tersebut akan mengembalikan variabel Luas. Apabila dalam program utama kita panggil x = luas_segitiga(2,3), maka nilai x adalah 3.

PROSEDUR

Prosedur pada dasarnya sama dengan fungsi, namun ia tidak mengembalikan hasil. Contoh :

```
1. def luas_segitiga(alas, tinggi):
2.       Luas = 0.5 * alas * tinggi
3.       print(Luas)
```

Prosedur luas_segitiga di atas adalah prosedur untuk menghitung luas segitiga. Parameternya adalah alas dan tinggi. Prosedur tersebut akan menuliskan luas dari segitiga berdasarkan parameter. Apabila dalam program utama kita panggil luas_segitiga(2,3), maka akan menuliskan luas segitiga tersebut yaitu 3.

Dipindai dengan CamScanner

SOAL-SOAL

1. Buatlah sebuah fungsi untuk menghitung luas segitiga. Lalu buat program yang menerima bilangan bulat A dan T. Gunakan fungsi untuk menghitung luas segitiga dengan parameter A dan T.
Contoh :

Input	Output
Masukkan alas segitiga : 2 Masukkan tinggi segitiga: 3	Luas segitiga adalah : 3.0

2. Buatlah program untuk menerima N, array A dan array T dengan panjang N. Lalu buatlah array L. Elemen L[i] adalah luas dari segitiga dengan alas A[i] dan tinggi T[i]. Hitung luas dengan membuat fungsi luas segitiga seperti No. 1. Setelah itu, keluarkan array L.
Contoh :

Input	Output
Masukkan banyak segitiga: 2 Masukkan alas segitiga ke-1: 2 Masukkan tinggi segitiga ke-1: 3 Masukkan alas segitiga ke-2: 2 Masukkan tinggi segitiga ke-2: 4	[3.0, 4.0]

3. Buatlah sebuah fungsi untuk menghasilkan nilai faktorial. Lalu buat program utama yang menerima A dan N. Tuliskan nilai dari faktorial A hingga faktorial N. Contoh :

Input	Output
Masukkan nilai A: 2 Masukkan nilai N: 5	Nilai 2! adalah : 2 Nilai 3! adalah : 6 Nilai 4! adalah : 24 Nilai 5! adalah : 120

4. Buatlah fungsi yang memiliki parameter A, B dan C, lalu kembalikan nilai terbesar dari keduanya. Lalu buat program utama untuk menerima A, B dan C. Gunakan fungsi untuk menuliskan nilai terbesar antara A, B dan C.
Contoh :

Input	Output
Masukkan nilai A: 3 Masukkan nilai B: 7 Masukkan nilai C: 2	Nilai terbesar adalah: 7

5. Buatlah fungsi untuk menerima sebuah string, lalu mengembalikan kebalikan dari string tersebut. Buatlah program utama yang menerima array A dari string dengan panjang N, lalu buatlah array B dari string. Elemen B[i] adalah string kebalikan dari A[i]. Tuliskan array A dan B.
Contoh :

Input	Output
Masukkan jumlah string: 3 Masukkan string ke-1: grogu Masukkan string ke-2: pengkom Masukkan string ke-3: jedi	Array A: ['grogu', 'pengkom', 'jedi'] Array B: ['ugorg', 'mokgnep', 'idej']

Dipindai dengan CamScanner

6. Buatlah fungsi dengan parameter N, dan periksa apakah N bilangan prima atau bukan. Fungsi akan mengembalikan bool (True/False). Lalu buat program utama yang menerima A dan B, lalu cek ada berapa bilangan prima pada rentang A hingga B.

Contoh :

Input	Output
Masukkan nilai A: 5 Masukkan nilai B: 15	Jumlah bilangan prima antara 5 dan 15 adalah : 4

7. **(UAS 2021)** Buatlah sebuah fungsi dengan parameter x dan n untuk mengembalikan hasil dari f(x) berikut :

Ωώ ρ ώ ώ ώ Ε ώ

Lalu buat program utama yang menerima x dan n, lalu keluarkan hasil f(x) dengan fungsi tersebut.

Contoh :

Input	Output
Masukkan nilai x: 2 Masukkan nilai n: 8	Hasil dari f(x) adalah 511

8. **(UAS 2021)** Sebuah formula Mean Squared Error (MSE) adalah sebagai berikut :

ü ΥΟ $\frac{\rho}{\varepsilon}$ ϋ ϋ

Buatlah fungsi untuk mencari nilai MSE. Lalu buatlah program utama yang menerima bilangan bulat n, lalu membuat array X dan Y dari float dengan panjang n. Hitung nilai MSE dan keluarkan hasilnya.

Contoh :

Input	Output
Masukkan nilai n: 2 Masukkan nilai X: 1 2 Masukkan nilai Y: 3 4	Nilai MSE adalah 4.0

9. Fungsi f(x) bernilai Ωώ ϋώ ώφ sedangkan fungsi g(x) bernilai Ωώ ϋώ ϋώ. Buatlah fungsi f(x), g(x), dan fungsi komposisi (gof)(x). Buat program utama yang menerima a,b, dan x. Hitung nilai (gof)(x) dengan fungsi yang telah dibuat.

Contoh :

Input	Output
Masukkan nilai a: 3 Masukkan nilai b: 2 Masukkan nilai x: 3	Nilai dari (gof)(x) : 75141

Dipindai dengan CamScanner

10. Buatlah sebuah fungsi yang menerima parameter string berformat waktu "JJ:MM:DD", lalu mengembalikan sebuah integer yang merupakan total detik dari waktu tersebut. Lalu buat program utama yang menerima N buah string sesuai format, lalu mengeluarkan total detik sesuai urutan.

Contoh :

Input	Output
Masukkan jumlah waktu: 3 Masukkan waktu ke-1: 1:23:45 Masukkan waktu ke-2: 2:34:56 Masukkan waktu ke-3: 1:22:21	Total detik waktu ke1 adalah: 5025 Total detik waktu ke2 adalah: 9296 Total detik waktu ke3 adalah: 4941

11. Buatlah sebuah prosedur untuk menuliskan "Hello, World!" sebanyak N kali. Lalu buat program utama yang menerima N, lalu menuliskan Hello World dengan prosedur tersebut.

Contoh :

Input	Output
Masukkan nilai N: 3	Hello, World! Hello, World! Hello, World!

12. Buatlah prosedur untuk membuat daftar menu makanan dalam sebuah kantin. Ikuti format dalam contoh di bawah :

Kantin (NAMA)
1) (Makanan 1)
2) (Makanan 2)
...

Lalu buat program utama yang menerima nama kantin, jumlah makanan (N), dan N baris berikutnya adalah makanan yang tersedia.

Contoh :

Input	Output
Masukkan nama kantin: Kwak Masukkan jumlah makanan yang tersedia: 3 Masukkan makanan ke-1: Nasi Goreng Masukkan makanan ke-2: Mie Goreng Masukkan makanan ke-3: Pecel Lele	##### Kantin Kwak ##### 1) Nasi Goreng 2) Mie Goreng 3) Pecel Lele

13. Ulangi soal No.11, buat program terus berjalan hingga user memasukkan N<0.

Contoh :

Input / Output
Masukkan nilai N: 2 Hello, World! Hello, World! Masukkan nilai N: 1 Hello, World! Masukkan nilai N: -1

14. Buatlah program utama yang menerima M, lalu menerima nama kantin, jumlah makanan, dan makanan yang tersedia (seperti nomor 12) sebanyak M kali. Tulis menu M kantin tersebut menggunakan prosedur seperti nomor 12.

Dipindai dengan CamScanner

Contoh :

Input	Output
Masukkan jumlah kantin : 2 Masukkan nama kantin ke-1: Kwak Masukkan jumlah makanan yang tersedia: 3 Masukkan makanan ke-1: Nasi Goreng Masukkan makanan ke-2: Mie Goreng Masukkan makanan ke-3: Pecel Lele Masukkan nama kantin ke-2: Kwik Masukkan jumlah makanan yang tersedia: 2 Masukkan makanan ke-1: Bakso Masukkan makanan ke-2: Mie Ayam	##### Kantin Kwak ##### 1) Nasi Goreng 2) Mie Goreng 3) Pecel Lele ##### Kantin Kwik ##### 1) Bakso 2) Mie Ayam

15. Buatlah prosedur untuk mengeluarkan teks perkenalan dengan format seperti di bawah :
"Halo, nama saya (NAMA). Saya adalah seorang (PEKERJAAN). Salam kenal!"
Lalu buat program utama yang menerima bilangan bulat N sebagai banyak orang, lalu menerima array NAMA dan PEKERJAAN dengan panjang N. Setelah itu tuliskan teks perkenalan menggunakan prosedur.
Contoh :

Input	Output
Masukkan jumlah orang: 3 Masukkan nama orang ke-1: Kwak Masukkan pekerjaan orang ke-1: Petani Masukkan nama orang ke-2: Kwik Masukkan pekerjaan orang ke-2: Insinyur Masukkan nama orang ke-3: Kwek Masukkan pekerjaan orang ke-3: Kang ojol	Halo, nama saya Kwak. Saya adalah seorang Petani. Salam kenal! Halo, nama saya Kwik. Saya adalah seorang Insinyur. Salam kenal! Halo, nama saya Kwek. Saya adalah seorang Kang ojol. Salam kenal!

Dipindai dengan CamScanner

PEMBAHASAN

1. Rumus dari luas segitiga adalah Ü -ßßY, dengan L adalah luas, A adalah alas, dan T adalah tinggi. Oleh karena itu, kita perlu dua parameter yaitu A dan T.

```
1. # Fungsi luas segitiga
2. def luas_segitiga(A,T):
3.     L = 0.5 * A * T
4.     return L
5.
6. # Program utama
7. A = float(input("Masukkan alas segitiga : "))
8. T = float(input("Masukkan tinggi segitiga: "))
9. print(f"Luas segitiga adalah : {luas_segitiga(A,T)}")
```

2. Karena kita akan mencari luas segitiga dari sekumpulan data array A dan array T, lakukan penginputan untuk kedua array tersebut. Lalu buat pengulangan untuk menggunakan fungsi yang telah dibuat pada nomor 1.

```
1. # Fungsi luas segitiga
2. def luas_segitiga(A,T):
3.     L = 0.5 * A * T
4.     return L
5.
6. # Program utama
7. N = int(input("Masukkan banyak segitiga: "))
8. # Inisiasi array A,T, dan L
9. A = [0.0 for i in range(N)]
10. T = [0.0 for i in range(N)]
11. L = [0.0 for i in range(N)]
12. # Input nilai alas dan tinggi dalam array
13. for i in range(N):
14.     A[i] = float(input(f"Masukkan alas segitiga ke-{i+1}: "))
15.     T[i] = float(input(f"Masukkan tinggi segitiga ke-{i+1}: "))
16. # Hitung luas dan masukkan ke array L
17. for i in range(N):
18.     L[i] = luas_segitiga(A[i],T[i])
19. # Tuliskan array L
20. print(L)
```

3. Nilai dari X! (X faktorial) adalah ρ z ς z σ z 8 z ώ. Maka kita buat fungsi pengulangan untuk mengalikan dari 1 hingga x. Setelah itu, pada program utama kita buat pengulangan dari A hingga N. Untuk setiap i, kita cari nilai faktorialnya menggunakan fungsi yang telah dibuat.

```
1. def faktorial(x):
2.     result = 1
3.     for i in range(1, x+1):
4.         result *= i
5.     return result
6. A = int(input("Masukkan nilai A: "))
```


Dipindai dengan CamScanner

```
7. N = int(input("Masukkan nilai N: "))
8. for i in range(A,N+1):
9.     print(f"Nilai {i}! adalah : {faktorial(i)}")
```

4. Ingat kembali soal pada bab BAB IV. Untuk menentukan nilai terbesar, kita gunakan perbandingan percabangan.

```
1. # Fungsi cari terbesar
2. def cari_terbesar(A, B, C):
3.     if A >= B and A >= C:
4.         return A
5.     elif B >= A and B >= C:
6.         return B
7.     else:
8.         return C
9.
10. # Program Utama
11. A = int(input("Masukkan nilai A: "))
12. B = int(input("Masukkan nilai B: "))
13. C = int(input("Masukkan nilai C: "))
14. terbesar = cari_terbesar(A, B, C)
15. print(f"Nilai terbesar adalah: {terbesar}")
```

5. Ingat bahwasanya string adalah array of char. Untuk membuat fungsi membalikkan string, kita buat string kosong s_balik. Lalu lakukan iterasi dari 0 hingga N-1. Ingat bahwa array dimulai dari 0. Pada setiap iterasi, fungsi mengambil karakter ke-i dari s dengan menggunakan indeks (N-1)-i. Ini dilakukan untuk memastikan bahwa karakter terakhir pada argumen s menjadi karakter pertama pada s_balik. Karakter yang diambil kemudian ditambahkan pada variabel "s_balik".

```
1. def balik(s):
2.     N = len(s)
3.     s_balik = ""
4.     for i in range(N):
5.         s_balik += s[(N-1)-i]
6.     return s_balik
7. N = int(input("Masukkan jumlah string: "))
8. A = [""] for i in range(N)]
9. B = [""] for i in range(N)]
10.
11. for i in range(N):
12.     A[i] = input(f"Masukkan string ke-{i+1}: ")
13.     B[i] = balik(A[i])
14. print(f"Array A: {A}")
15. print(f"Array B: {B}")
```

6. Bilangan prima adalah bilangan yang hanya habis dibagi dengan 1 dan dengan angka itu sendiri. Untuk mengecek apakah N bilangan prima, cara termudahnya adalah kita hitung ada berapa angka dari 1 hingga N yang habis membagi N. Apabila hanya ada 2, maka dipastikan N adalah bilangan prima. Pada program utama kita lakukan

Dipindai dengan CamScanner

loop dari A hingga B. Lalu kita gunakan fungsi yang telah dibuat untuk mengecek apakah i adalah bilangan prima.

```
def prima(n):
    total = 0
    for i in range(1,n+1):
        if (n%i==0):
            total+=1
    if(total==2):
        return True
    else:
        return False

# Program Utama
A = int(input("Masukkan nilai A: "))
B = int(input("Masukkan nilai B: "))

count = 0
for i in range(A, B+1):
    if prima(i):
        count += 1

print(f"Jumlah bilangan prima antara {A} dan {B} adalah : {count}")
```

7. Kasus ini mirip dengan soal No.3. Namun ini adalah penjumlahan, bukan perkalian.

```
def f(x, n):
    result = 0
    for i in range(n+1):
        result += x**i
    return result

# Program Utama
x = int(input("Masukkan nilai x: "))
n = int(input("Masukkan nilai n: "))

print("Hasil dari f(x) adalah", f(x, n))
```

8. Untuk mengimplementasikan sigma B ΩΩ dalam python, pertama kita inisiasi variabel untuk melakukan penjumlahan. Setelah itu buat loop dari x hingga n, lalu variabel penjumlah itu ditambah dengan nilai f(i).

```
def mse(n, X, Y):
    result = 0
    for i in range(n):
        result += (X[i] - Y[i])**2
    result = result * (1/n)
    return result

# Program Utama
n = int(input("Masukkan nilai n: "))
X = [0.0 for i in range(n)]
```


Dipindai dengan CamScanner

```
Y = [0.0 for i in range(n)]

# Masukkan elemen array X
print("Masukkan nilai X:")
for i in range(n):
    X[i] = int(input())

# Masukkan elemen array Y
print("Masukkan nilai Y:")
for i in range(n):
    Y[i] = int(input())

print("Nilai MSE adalah", mse(n, X, Y))
```

9. Fungsi komposisi (gof)(x) pada dasarnya adalah memasukkan nilai f(x) pada g(x), atau dapat ditulis g(f(x)). Pada soal ini kita membutuhkan parameter tambahan a dan b. Kita buat dan implementasikan kedua fungsi f(x,a,b) dan g(x,a,b) dalam python. Lalu panggil g(x,a,b), dan untuk mengomposisikannya, ganti x dengan f(x,a,b) sehingga menjadi g(f(x,a,b),a,b).

```
def f(x,a,b):
    result = a*(x**2)+b*x
    return result
def g(x,a,b):
    result = b*(x**3)+a*(x**2)
    return result

a = int(input("Masukkan nilai a: "))
b = int(input("Masukkan nilai b: "))
x = int(input("Masukkan nilai x: "))
gof = g(f(x,a,b),a,b)

print(f"Nilai dari (gof)(x) : {gof}")
```

10. Pada fungsi ini kita dapat menggunakan string split untuk memisahkan string terhadap karakter tertentu. Misal string s = "JJ:MM:DD". Kita lakukan split dengan x = s.split(":"). Maka x adalah array of string yang bernilai ["JJ", "MM", "DD"]. Untuk mencari total detik, maka kita konversi string tersebut menjadi int dan kalikan dengan nilai detiknya.

```
def jam_ke_detik(str_waktu):
    jam = str_waktu.split(":")
    total_detik = (int(jam[0])*3600) + (int(jam[1])*60) + int(jam[2])
    return total_detik
n = int(input("Masukkan jumlah waktu: "))
waktu = ["" for i in range(n)]
for i in range(n):
    waktu[i] = input(f"Masukkan waktu ke-{i+1}: ")
for i in range(n):
    detik = jam_ke_detik(waktu[i])
```


Dipindai dengan CamScanner

```
    print(f"Total detik waktu ke{i+1} adalah: {detik}")
```

11. Gunakan pengulangan sebanyak n kali untuk menulis Hello World sebanyak n kali.

```
def print_hello_world(n):
    for i in range(n):
        print("Hello, World!")

# Program Utama
n = int(input("Masukkan nilai N: "))
print_hello_world(n)
```

12. Prosedur yang akan dibuat membutuhkan dua parameter, yakni nama kantin (string) dan makanan (array of string). Untuk menulis list makanan, gunakan loop sebanyak banyaknya makanan, lalu tuliskan tiap makanannya.

```
def print_menu(nama_kantin, makanan):
    print(f"##### Kantin {nama_kantin} #####")
    for i in range(len(makanan)):
        print(f"{i+1}) {makanan[i]}")

# Program Utama
nama_kantin = input("Masukkan nama kantin: ")
n = int(input("Masukkan jumlah makanan yang tersedia: "))
makanan = [""" for i in range(n)]
for i in range(n):
    makanan[i] = input(f"Masukkan makanan ke-{i+1}: ")
print_menu(nama_kantin,makanan)
```

13. Gunakan loop while untuk memeriksa. Selama n tidak negatif, maka penulisan Hello World dan memasukkan n baru tetap berjalan.

```
# Fungsi menuliskan hello wolrd sebanyak n
def print_hello_world(n):
    for i in range(n):
        print("Hello, World!")

# Program utama
n = int(input("Masukkan nilai N: "))
# Selama n>0
while n >= 0:
    print_hello_world(n)
    n = int(input("Masukkan nilai N: "))
```

14. Soal ini mirip soal No. 12. Kita perlu merombak program utama kita. Nama kantin yang awalnya string kita ubah jadi array of string. Daftar makanan yang awalnya array of string kita ubah jadi array dalam array (matriks). Tiap nama kantin kita masukkan dalam array nama_kantin, dan daftar makanan tiap kantin kita masukkan dalam array menu. Panjang array dari daftar makanan adalah tidak menentu. Kita tidak bisa menginisiasi array menu terlebih dahulu. Sehingga kita buat array kosong, dan untuk menambahkan daftar makanan pada array menu, kita gunakan append.

Dipindai dengan CamScanner

```
def print_menu(nama_kantin, makanan):
    print(f"##### Kantin {nama_kantin} #####")
    for i in range(len(makanan)):
        print(f"{i+1}) {makanan[i]}")

# Program Utama
M = int(input("Masukkan jumlah kantin : "))
nama_kantin = [""" for i in range(M)]
menu = []
# Memasukkan data setiap kantin
for i in range(M):
    nama_kantin[i] = input(f"Masukkan nama kantin ke-{i+1}: ")
    n = int(input("Masukkan jumlah makanan yang tersedia: "))
    makanan = [""" for i in range(n)]
    # Masukkan makanan kantin ke-i
    for j in range(n):
        makanan[j] = input(f"Masukkan makanan ke-{j+1}: ")
    menu.append(makanan)
# Tuliskan menu tiap kantin
for i in range(M):
    print_menu(nama_kantin[i], menu[i])
    print()
```

15. Prosedur langsung kita buat untuk mengimplementasikan template perkenalan. Dalam program utama kita menyimpan nama-nama dan pekerjaan setiap orang ke dalam array. Lalu buat loop untuk menuliskan template perkenalan menggunakan prosedur.

```
def perkenalan(nama, pekerjaan):
    print(f"Halo, nama saya {nama}. Saya adalah seorang {pekerjaan}.
    Salam kenal!")

n = int(input("Masukkan jumlah orang: "))
nama = [""" for i in range(n)]
pekerjaan = [""" for i in range(n)]

for i in range(n):
    nama[i] = input(f"Masukkan nama orang ke-{i+1}: ")
    pekerjaan[i] = input(f"Masukkan pekerjaan orang ke-{i+1}: ")

for i in range(n):
    perkenalan(nama[i], pekerjaan[i])
    print()
```


Dipindai dengan CamScanner

BAB VIII: Pengenalan AI

AI adalah kependekan dari Artificial Intelligence, yaitu kemampuan mesin atau komputer untuk meniru kemampuan kognitif manusia. Teknologi-teknologi seperti machine learning, neural networks, dan natural language processing digunakan untuk mengembangkan sistem atau program komputer yang dapat belajar dari data dan mengambil keputusan atau tindakan. AI dapat diterapkan pada berbagai bidang, seperti robotika, kendaraan otonom, kesehatan, pendidikan, dan bisnis. Pengujian AI adalah menggunakan turing test, yaitu sebuah uji coba untuk mengukur kemampuan sebuah mesin atau komputer untuk menunjukkan perilaku yang setara dengan perilaku manusia.

AI memiliki 4 pendekatan, yaitu :

- **Thinking humanly (berpikir seperti manusia):** memahami cara kerja otak manusia dan mengembangkan sistem AI yang dapat meniru cara manusia berpikir. Pendekatan ini mempelajari psikologi, linguistik, dan ilmu kognitif manusia.
- **Thinking rationally (berpikir secara rasional):** mengembangkan sistem AI yang dapat melakukan pemrosesan informasi dengan cara yang logis dan rasional. Pendekatan ini menggunakan matematika dan logika untuk mengembangkan algoritma dan aturan untuk mengambil keputusan.
- **Acting humanly (bertindak seperti manusia):** mengembangkan sistem AI yang dapat berinteraksi dengan manusia seperti halnya manusia berinteraksi dengan sesama manusia. Pendekatan ini menggunakan metode seperti pemrosesan bahasa alami, pengenalan wajah, dan analisis sentimen untuk menciptakan interaksi yang manusiawi.
- **Acting rationally (bertindak secara rasional):** mengembangkan sistem AI yang dapat bertindak secara optimal dalam lingkungan tertentu. Pendekatan ini menggunakan teori keputusan dan pemodelan untuk mengembangkan sistem yang dapat membuat keputusan cerdas berdasarkan data dan informasi yang tersedia.

Terdapat tiga tipe atau kelas umum dari AI, yaitu problem solving agent, knowledge-based agent, dan learning agent. Problem solving agent bertujuan untuk menyelesaikan suatu masalah atau tugas dengan menggunakan algoritma atau aturan yang telah diprogram sebelumnya. Knowledge-based agent didasarkan pada pengetahuan yang dimiliki oleh sistem, sedangkan learning agent dapat belajar dari data untuk memperbaiki performanya.

Dalam problem solving agent, ada dua algoritma yang digunakan yaitu BFS (Breadth-First Search) dan DFS (Depth-First Search), yang digunakan dalam AI untuk melakukan pencarian ruang keadaan atau graf. BFS mencari solusi dengan mengekspansi simpul-simpul terdekat terlebih dahulu, sedangkan DFS mencari solusi dengan mengeksplorasi simpul yang paling dalam terlebih dahulu. Pemilihan algoritma yang tepat tergantung pada masalah yang ingin dipecahkan.

Dipindai dengan CamScanner

SOAL-SOAL

1. Jelaskan mengenai pengertian AI
2. Jelaskan 4 pendekatan AI
3. (UAS 2021) Berikan dan jelaskan contoh penggunaan AI dalam bidang pendidikan
4. Berikan dan jelaskan contoh penggunaan AI dalam bidang robotika
5. Berikan dan jelaskan contoh penggunaan AI dalam bidang kesehatan
6. Berikan dan jelaskan contoh penggunaan AI dalam bidang bisnis
7. Berikan dan jelaskan contoh penggunaan AI dalam bidang transportasi
8. Berikan dan jelaskan contoh penggunaan AI dalam bidang seni
9. Jelaskan tiga tipe AI
10. Jelaskan dua algoritma dalam AI tipe problem solving agent
11. Diketahui sebuah 8-puzzle yang memuat 8 buah ubin yang diberi nomor 1 sampai 8, dan satu buah slot kosong yang digunakan untuk menggerakan ubin. Keadaan awal dari puzzle ini adalah :

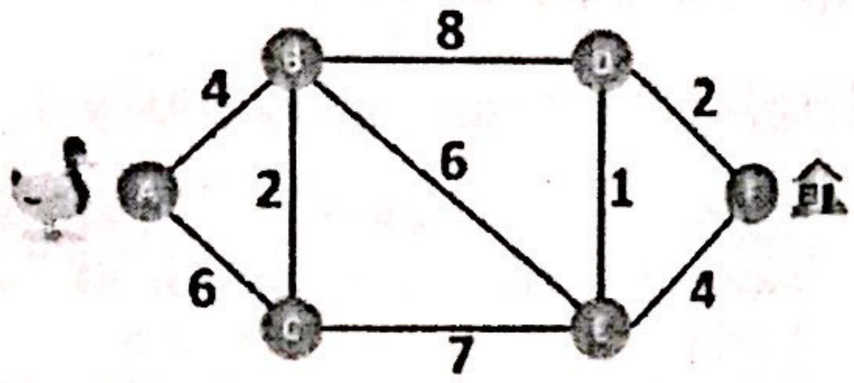
1	2	3
6	4	5
7	8	

Sedangkan untuk keadaan akhir yang diharapkan adalah :

1	2	3
4		5
6	7	8

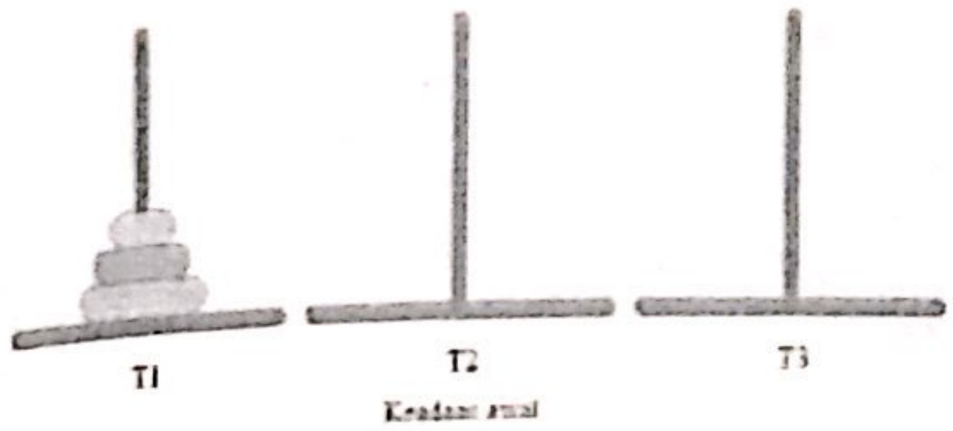
a. Tuliskan/gambarkan 1 (satu) alternatif langkah-langkah untuk mencapai keadaan akhir. Tuliskan berapa langkah yang Anda butuhkan untuk mencapai hal tersebut!
b. Jelaskan ide untuk dapat menemukan langkah yang Anda ilustrasikan. Anda dapat menggunakan pendekatan BFS, DFS, atau pendekatan lain.

12. Kwak tersesat di kota A. Ia lalu membuka Maps untuk mencari jalan pulang. Anda adalah algoritma dari aplikasi Maps tersebut. Carilah jalan terpendek antara kota A hingga kota F (rumah Kwak). Peta digambarkan dalam graf berikut :

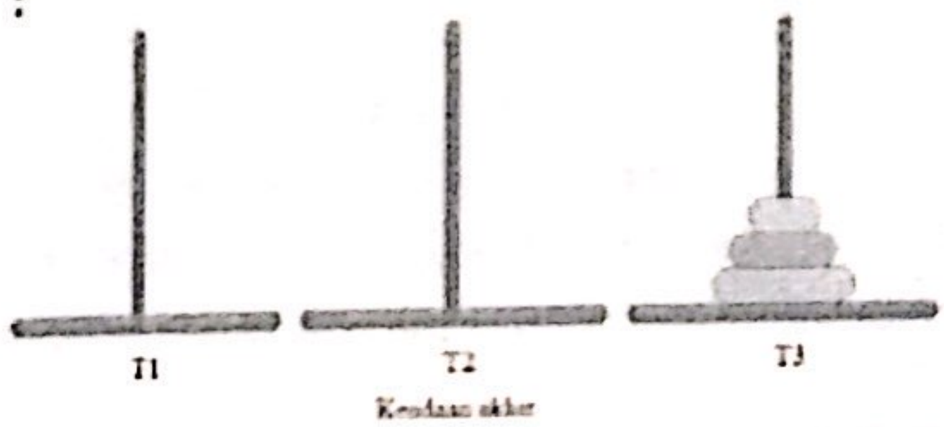


a. Tuliskan jalan alternatif terpendek untuk Kwak mencapai kota F. Tuliskan total jarak yang ditempuh oleh Kwak.
b. Jelaskan ide untuk menemukan langkah yang Anda ilustrasikan.

13. (UAS 2021) Diketahui 3 (tiga) buah tiang T1, T2 dan T3. Pada keadaan awal tiang T1 berisi 3 (tiga) buah cakram yang ditumpuk terurut berdasarkan ukuran cakram (cakram terkecil paling atas dan terbesar di paling bawah) sebagaimana dapat dilihat pada gambar di bawah ini :



Tujuan dari permainan ini adalah memindahkan seluruh cakram dari T1 ke T3, dengan cakram tersusun dengan urutan yang sama untuk mencapai keadaan akhir yang dapat dilihat pada gambar di bawah ini :



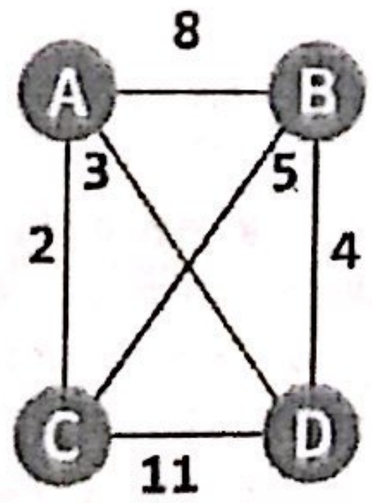
Syarat dari perpindahan adalah sebagai berikut :

- 1 (satu) langkah perpindahan hanya dapat memindahkan 1 (satu) buah cakram dari satu tiang ke tiang lain
- Cakram hanya dapat dipindahkan ke tiang yang kosong atau jika tiang tidak kosong, cakram hanya dapat diletakkan di atas cakram lain yang lebih besar.

a. Tuliskan/gambarkan 1 (satu) alternatif langkah-langkah keadaan awal sehingga mencapai keadaan akhir yang diharapkan. Tuliskan berapa banyak langkah yang Anda butuhkan untuk mencapai hal tersebut.

b. Jelaskan ide untuk dapat menemukan langkah yang Anda ilustrasikan. Anda dapat menggunakan pendekatan BFS, DFS, atau pendekatan lain.

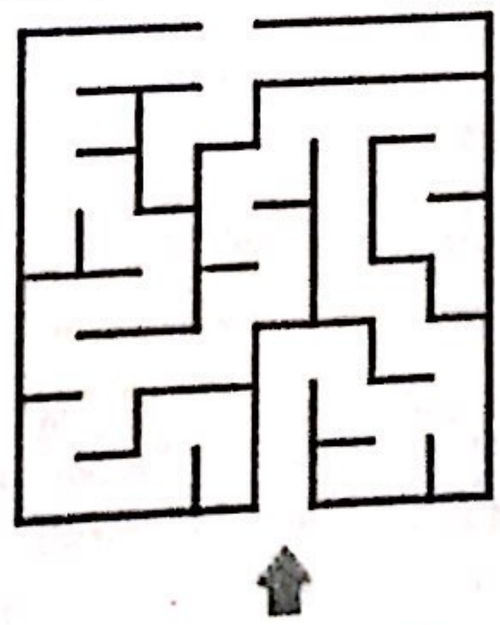
14. Kwik sedang menjadi tukang pos. Ia harus mengirim surat ke desa A, B, C, dan D. Kantor pos terletak di desa A. Ia berangkat dari kota A dan harus kembali ke kota A. Peta antar desa digambarkan dalam graf berikut :



a. Tuliskan alternatif jalan terpendek untuk Kwik mengantarkan seluruh surat ke 4 desa tersebut.

b. Jelaskan ide untuk dapat menemukan langkah yang Anda tuliskan. Anda dapat menggunakan pendekatan BFS, DFS, atau pendekatan la

15. Sebuah maze diilustrasikan pada gambar berikut :



a. Gambarkan alternatif penyelesaian dari maze di atas. Tuliskan tahapan-tahapan yang dilalui.

b. Jelaskan ide untuk dapat menemukan langkah yang Anda tuliskan. Anda dapat menggunakan pendekatan BFS, DFS atau pendekatan lain.

PEMBAHASAN

1. AI adalah kependekan dari Artificial Intelligence, yaitu kemampuan mesin atau komputer untuk meniru kemampuan kognitif manusia. Teknologi-teknologi seperti machine learning, neural networks, dan natural language processing digunakan untuk mengembangkan sistem atau program komputer yang dapat belajar dari data dan mengambil keputusan atau tindakan. AI dapat diterapkan pada berbagai bidang, seperti robotika, kendaraan otonom, kesehatan, pendidikan, dan bisnis.

2. Pendekatan AI yang umum digunakan meliputi empat kategori, yaitu thinking humanly, thinking rationally, acting humanly, dan acting rationally.
   - Thinking humanly (berpikir seperti manusia): pendekatan ini berusaha untuk memahami cara kerja otak manusia dan mengembangkan sistem AI yang dapat meniru cara manusia berpikir. Pendekatan ini mempelajari psikologi, linguistik, dan ilmu kognitif manusia.
   - Thinking rationally (berpikir secara rasional): pendekatan ini berusaha untuk mengembangkan sistem AI yang dapat melakukan pemrosesan informasi dengan cara yang logis dan rasional. Pendekatan ini menggunakan matematika dan logika untuk mengembangkan algoritma dan aturan untuk mengambil keputusan.
   - Acting humanly (bertindak seperti manusia): pendekatan ini berusaha untuk mengembangkan sistem AI yang dapat berinteraksi dengan manusia seperti halnya manusia berinteraksi dengan sesama manusia. Pendekatan ini menggunakan metode seperti pemrosesan bahasa alami, pengenalan wajah, dan analisis sentimen untuk menciptakan interaksi yang manusiawi.
   - Acting rationally (bertindak secara rasional): pendekatan ini berusaha untuk mengembangkan sistem AI yang dapat bertindak secara optimal dalam lingkungan tertentu. Pendekatan ini menggunakan teori keputusan dan pemodelan untuk mengembangkan sistem yang dapat membuat keputusan cerdas berdasarkan data dan informasi yang tersedia.

3. Beberapa contoh AI dalam bidang pendidikan adalah :
   - Smart Content : AI digunakan untuk menyesuaikan konten berdasarkan kemampuan siswanya. Misalnya dalam Coursera dan Duolingo.
   - Interactive Tutoring : AI dapat membantu mengajarkan dan menjawab pertanyaan siswa secara real time. Misalnya ChatGPT.
   - Adaptive Learning Environment : AI dapat mempersonalisasi lingkungan belajar siswa secara khusus agar siswa lebih nyaman dalam belajar.

4. Beberapa contoh Ai dalam bidang robotika adalah :
   - Pengenalan wajah : Teknologi AI digunakan dalam robotika untuk memungkinkan robot mengenali wajah manusia dan berinteraksi dengan mereka. Misalnya Face ID dalam Apple iPhone, dan Face Recognition dalam Android.
   - Asisten pribadi : AI digunakan untuk membantu manusia dalam kegiatan sehari-hari serta memberi informasi. Misalnya Google Assistant, Siri, Bixby, Amazon Alexa, dan sebagainya.
   - Kendali gerakan : AI dapat digunakan untuk mengendalikan gerakan robot, misalnya pergerakan kaki, lengan, dan lain sebagainya. Misalnya robot karya Boston Dynamic.

Dipindai dengan CamScanner

5. Beberapa contoh AI dalam bidang kesehatan adalah :
   - Diagnosis penyakit: AI digunakan untuk mendiagnosis penyakit lebih cepat dan akurat dengan menganalisis data medis menggunakan machine learning.
   - Prediksi risiko kesehatan: AI digunakan untuk mengidentifikasi risiko kesehatan pada pasien berdasarkan data medis, memungkinkan untuk melakukan tindakan pencegahan lebih awal.
   - Asisten medis: AI digunakan untuk mengembangkan asisten medis yang membantu dokter dalam menentukan rencana perawatan pasien, memberikan rekomendasi terapi, atau memantau kondisi pasien secara real-time.
6. Beberapa cnotoh AI dalam bidang bisnis adalah :
   - Pemasaran : AI digunakan untuk menganalisis data konsumen dan memberi rekomendasi pemasaran yang tepat.
   - Customer Service : AI digunakan untuk memproses permintaan pelanggan dan memberikan respons yang cepat dan akurat. Misalnya Veronika dari Telkomsel.
   - Analisis data : AI digunakan untuk menganalisis data bisnis besar dan kompleks dan memberikan informasi yang berharga tentang tren pasar, preferensi konsumen, dan peluang bisnis baru.
7. Beberapa contoh AI dalam bidang transportasi adalah :
   - Kendaraan Autonomus : AI digunakan dalam kendaraan untuk dapat mengemudi sendiri tanpa bantuan manusia. Misalnya Tesla Autopilot, Hyundai Line Keep Assist, dan sebagainya.
   - Pengatur lalu lintas : AI digunakan untuk mengatur lalu lintas jalan raya dan memantau secara real time, menyesuaikan waktu lampu merah dan hijau dalam lalu lintas.
   - Navigasi : AI digunakan untuk menemukan jalan terpendek dan tercepat untuk kita saat berpergian. Misalnya Google Maps, Waze, dan sebagainya.
8. Beberapa contoh AI dalam bidang seni adalah :
   - Generative AI : AI digunakan untuk membuat karya seni generatif yang unik dan bervariasi. Misalnya DALL-E untuk membuat seni rupa sesuai teks yang kita masukkan.
   - Fotografi Adaptif : AI digunakan dalam kamera-kamera smartphone untuk menyesuaikan objek dan lingkungan yang akan difoto. Misalnya pada brand smartphone Huawei, Oppo, Vivo.
   - Seni Interaktif : AI digunakan untuk membuat manusia dapat berinteraksi dengan karya seninya sendiri. Misalnya pada aplikasi TikTok.
9. Tiga tipe AI yang umum digunakan adalah :
   - Problem Solving Agent : Yaitu AI yang dirancang untuk menyelesaikan masalah atau tugas tertentu. Misalnya pada penerjemahan antar bahasa seperti Google Translate.
   - Knowledge-Based Agent : Yaitu AI yang telah diberikan pengetahuan sebelumnya sebelum melaksanakan tugas-tugasnya. Pengetahuan ini diberikan untuk membuat keputusan dan solusi yang dihadapi. Contoh dari tipe ini adalah ChatGPT yang diberikan banyak pengetahuan, namun diputus hingga September 2021.
   - Learning Agent : AI tipe ini memiliki kemampuan untuk belajar dari pengalaman dengan data-data baru yang ia terima.

Dipindai dengan CamScanner

10. Ada dua algoritma yang sering digunakan dalam AI Problem Solving Agent.
    - Breadth-First Search (BFS), yaitu mencari solusi dengan mengeksplor simpul-simpul yang lebih dangkal dahulu sebelum memeriksa simpul yang lebih dalam.
    - Depth-First Search (DFS), yaitu mencari solusi dengan mengeksplor simpul hingga dalam sebelum lanjut ke cabang simpul lainnya.

11. Dalam pembahasan ini kita gunakan metode BFS. Untuk menyelesaikannya, pertama kita tetapkan urutan operator yang akan digunakan. Misalnya adalah Left (L), Up (U), Right (R), dan Down (D).
    a. Scan QR Code berikut untuk penjelasan video animasi:

    Alternatif hasil yang diperoleh adalah : Ü° Ü° ˜Y° ˜Y
    Dengan banyak langkah : 4
    b. Metode yang digunakan adalah metode BFS (Breadth-First Search), yaitu mencari solusi dengan mengeksplor simpul-simpul yang lebih dangkal dahulu sebelum memeriksa simpul yang lebih dalam.

12. Untuk menyelesaikannya, kita dapat menggunakan cara berpikir algoritma dijkstra.
    a. Scan QR Code berikut untuk penjelasan video animasi:

    Alternatif hasil yang diperoleh adalah : ö° ö° ˜O° ˜O° ˜O
    Dengan jarak : 13 satuan
    b. Metode yang digunakan adalah menggunakan algoritma dijkstra. Algoritma Dijkstra adalah algoritma yang digunakan untuk menyelesaikan masalah jarak terpendek pada graf berbobot. Berikut adalah step-by-step dari algoritma Dijkstra:
        - Tentukan node (kota) awal dan akhir
        - Buat percabangan dari node ke node berikutnya secara berurutan
        - Catat jarak dari node ke node tetangganya, tandai node tersebut sebagai node yang telah diproses.
        - Apabila node tetangga sudah memiliki jarak, maka bandingkan jarak dari jalur yang sekarang dengan yang dihitung sebelumnya. Ambil jarak yang paling pendek
        - Eliminasi jalan yang tidak diperlukan
        - Jarak terpendek dari node awal hingga akhir adalah nilai jarak terakhir dari node akhir.

Dipindai dengan CamScanner

13. (UAS 2021) Dalam pembahasan ini kita akan menggunakan metode BFS. Untuk menyelesaikannya, pertama kita tetapkan urutan operator yang akan digunakan. Misalnya adalah cakram kecil ke tiang paling kiri, cakram sedang ke tiang paling kiri, cakram besar ke tiang paling kiri.

a. Scan QR Code berikut untuk penjelasan video animasi:

Alternatif solusi :

Langkah	Tindakan
Langkah 1	Cakram kecil ke T3
Langkah 2	Cakram sedang ke T2
Langkah 3	Cakram kecil ke T2
Langkah 4	Cakram besar ke T3
Langkah 5	Cakram kecil ke T1
Langkah 6	Cakram sedang ke T3
Langkah 7	Cakram kecil ke T3

b. Metode yang digunakan adalah metode BFS (Breadth-First Search), yaitu mencari solusi dengan mengeksplor simpul-simpul yang lebih dangkal dahulu sebelum memeriksa simpul yang lebih dalam.

14. Dalam pembahasan ini kita akan menggunakan metode Brute Force.

a. Untuk menyelesaikannya, carilah seluruh kemungkinan yang ada pada soal ini. Lalu, carilah jarak yang ditempuh oleh Kwik.

Kemungkinan	Urutan	Jarak
1	A-B-C-D-A	8 + 5 + 11 + 5 = 29
2	A-B-D-C-A	8 + 4 + 11 + 3 = 26
3	A-C-B-D-A	3 + 5 + 4 + 5 = 17
4	A-C-D-B-A	3 + 11 + 4 + 8 = 26
5	A-D-B-C-A	5 + 4 + 5 + 3 = 17
6	A-D-C-B-A	5 + 11 + 5 + 8 = 29

Ada dua alternatif jalan terpendek :

- A-C-B-D-A
- A-D-B-C-A

Kedua alternatif jalan tersebut memiliki jarak tempuh 17 satuan.

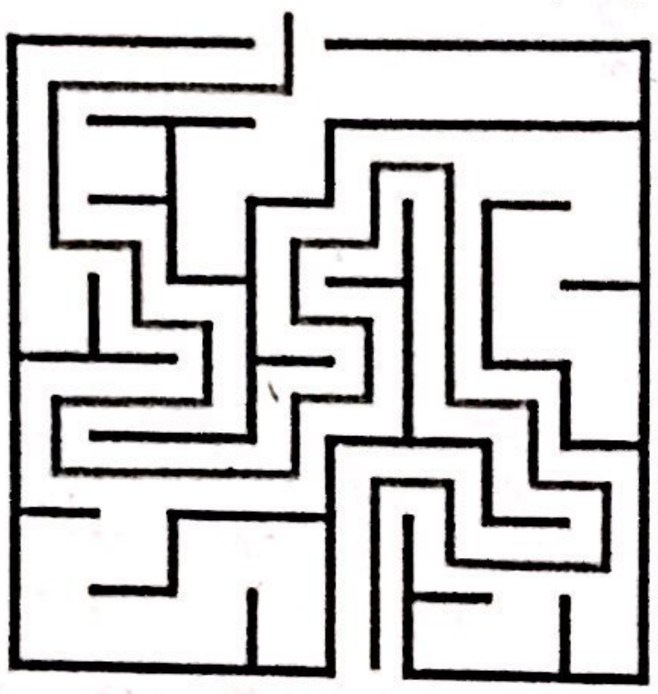
b. Metode yang digunakan adalah metode brute force. Brute force adalah metode penyelesaian masalah yang mencoba semua kemungkinan solusi secara sistematis dengan menguji masing-masing solusi satu per satu, sampai solusi yang benar ditemukan. Metode ini dapat diterapkan pada berbagai jenis masalah, termasuk masalah optimasi dan pemecahan masalah kombinatorial. Namun, metode brute force seringkali tidak efisien untuk masalah yang sangat kompleks karena memerlukan waktu yang sangat lama untuk mencoba semua kemungkinan solusi.

Dipindai dengan CamScanner

15. Dalam pembahasan ini kita akan menggunakan metode Depth-First Search (DFS).

a. Untuk menyelesaikannya, kita tentukan urutan operasi. Misalnya belok kiri, lurus, lalu belok kanan. Scan QR Code berikut untuk penjelasan video animasi:

Didapatkan alternatif penyelesaian berikut :



Dan didapatkan pula tahapan-tahapan penyelesaian :

ô ú ŵŷùû ó ì ŝ ŭ ŢP ŢP ŢP ŢP ó ì ŝ ŭ ŵè ŷèû ó ì ŝ ŭ ŢP ŢYŢà ŢI ŵŢ

b. Metode yang digunakan adalah metode DFS. Depth-First Search (DFS) adalah metode dimana kita mencari solusi dengan mengeksplor simpul hingga dalam sebelum lanjut ke cabang simpul lainnya. Pada soal ini, kita terus belok kiri hingga tidak ada opsi belok kiri. Jika sudah tidak ada, maka lurus. Jika sudah tidak ada opsi lurus, maka belok kanan.

Dipindai dengan CamScanner

BAB IX: Analisis Data

IMPORT DATA

Mengimport data excel :

```
1. import pandas as pd
2. df =
pd.read_excel("namafile.xlsx")
```

Mengimport data csv :

```
1. import pandas as pd
2. df = pd.read_csv("namafile.csv")
```

DATA CLEANING

- Menghapus data kosong :

```
1. df.dropna(inplace=True)
```

- Menghapus data duplikat :

```
1.
df.drop_duplicates(inplace=Tru
e)
```

- Mempositifkan data :

```
1. df['kolom'] =
df['kolom'].abs()
```

STATISTIKA

- Banyak data dataframe :

```
1. banyakdata = len(df)
```

- Banyak data kategorial :
  Misal kolom Gender terdapat nilai Pria dan Wanita

```
1. Jenis_gender =
df['Gender'].nunique()
```

  Jenis_gender akan menghasilkan : 2

- Deskripsi umum :

```
1. print(df.describe())
```

- Rata-rata :

```
1. rata = df['kolom'].mean()
```

- Standar deviasi :

```
1. deviasi = df['kolom'].std(
```

- Nilai maksimum :

```
1. maks = df['kolom'].max()
```

- Nilai minimum :

```
1. mins = df['kolom'].min()
```

MENGAKSES DATA

Untuk mengakses data dengan indeks, lakukan :

```
1. df[a,b]
```

dengan a adalah indeks awal dan b adalah indeks akhir. Misal :

- Mengambil data ke-4 :

```
1. df[4]
```

- Mengambil 10 data pertama :

```
1. df[:10]
```

- Mengambil data ke-10 sampai ke-20 :

```
1. df[9:19]
```

- Mengambil data ke-20 sampai terakhir :

```
1. df[20:]
```

Untuk mengakses data khusus, lakukan:

```
1. df.loc[(boolean)]
```

Pada (boolean) dapat diisi dengan pernyataan. Misalnya :

- Mengambil data dengan kolom Merk yang bernilai Eiger

```
1. data_eiger =
df.loc[df['Merk']== 'Eiger']
```

- Mengambil data dengan kolom Nomor_Sepatu yang bernilai di atas 40

```
1. data_up40 =
df.loc[df['Nomor_Sepatu']>40]
```

- Mengambil data dengan kolom Gender yang bernilai Wanita dan harga di atas 300000

```
1. data_sepatu_wanita_300 =
df.loc[(df['Gender'] ==
'Wanita') & (df['Harga'] >
300000)]
```

Untuk mendapatkan nilai (value) dari dataframe dengan banyak data 1, gunakan :

```
1. df_x['kolom'].values[0]
```

Contoh :

Dipindai dengan CamScanner

```
1. DataDengklek =
df.loc[df['Nama']== 'Dengklek']
2. Umur =
DataDengklek['Umur'].values[0]
3. print(f'Umur Pak Dengklek adalah
{Umur}')
```

FUNGSI-FUNGSI DATAFRAME

- Mencari nilai korelasi

```
1. Korelasi =
df['kolom1'].corr(df['kolom2'])
```

  - Bila nilai Korelasi mendekati -1, berarti berbanding terbalik
  - Bila nilai Korelasi mendekati 0, berarti tidak memiliki hubungan
  - Bila nilai Korelasi mendekati 1, berarti berbanding lurus

- Mengurutkan data dari yang terkecil

```
1. df.sort_values(['kolom'],
ascending=1)
```

- Mengurutkan data dari yang terbesar

```
1. df.sort_value(['kolom'],
ascending=0)
```

- Mengelompokkan data

```
1. df.groupby('kolom')
```

- Membuat dataframe tabel frekuensi suatu kolom

```
1. df.groupby('kolom').size()
```

- Menggabungkan dataframe df2 ke df1

```
1. df1 = pd.concat([df1, df2],
axis=1)
```


Dipindai dengan CamScanner

SOAL-SOAL

Untuk 5 soal berikutnya, silahkan scan QR Code berikut untuk mengunduh data :

Keterangan :

- Place_Id : nomor urut
- Place_Name : Nama tempat wisata
- Category : Kategori tempat wisata
- City : Lokasi kota tempat wisata
- Price : Harga tiket masuk (Jika 0, berarti gratis)
- Rating : Penilaian dari pengunjung, range 0 - 50

1. **(UAS 2021)** Pada data tersebut, terdapat beberapa data yang bernilai N/A ataupun negatif. Jika Anda diminta untuk melakukan analisis terhadap data tersebut, Apa yang Anda lakukan terhadap data-data tersebut?

2. Ambillah data-data berikut. Pertimbangkan juga dengan apa yang telah Anda lakukan pada No. 1!
   a. **(H.5 2021)** Banyak tempat wisata valid yang ada di data tersebut
   b. **(H.5 2021)** Tabel 10 data pertama
   c. **(H.5 2021)** Tabel Data ke-30 hingga 40
   d. **(H.5;P.5 2021)** Data tempat wisata Gedung Sate
   e. Berapa harga tiket masuk ke Gedung Sate?

3. Hitunglah statistik terhadap beberapa kolom berikut!
   a. **(UAS 2021)** Berapa rata-rata penilaian tempat wisata di Indonesia?
   b. **(UAS 2021)** Berapa standar deviasi harga tiket masuk wisata di Indonesia?
   c. **(UAS;H.5 2021)** Ada berapa tempat wisata di Bandung?
   d. **(P.5 2021)** Buatlah tabel untuk menyebutkan ada berapa tempat wisata di tiap kota!
   e. **(UAS;H.5;P.5 2021)** Berapa korelasi antara harga tiket masuk dengan rating tempat wisata? Jelaskan bagaimana korelasinya!

4. Jawablah pertanyaan-pertanyaan berikut dan buktikan dengan program :
   a. Dimana letak Situ Cileunca?
   b. Tempat wisata apa yang memiliki harga tiket paling mahal?
   c. Tempat wisata apa saja yang memiliki rating terbaik di Yogyakarta?
   d. Apakah Masjid Salman ITB berbayar? Jika iya, berapa harganya?
   e. Apa kategori tempat wisata termahal Bandung?

5. Buatlah tabel yang memuat data-data berikut :
   a. Tabel 5 tempat wisata dengan harga tiket masuk termurah
   b. **(P.5 2021)** Tabel 10 tempat wisata dengan harga tiket masuk termahal di Semarang
   c. Tabel Tempat wisata dengan rating di atas 45
   d. Tabel tempat wisata cagar alam yang ada di Yogyakarta
   e. **(P.5 2021)** Tabel jumlah tempat wisata gratis di masing-masing kota

Dipindai dengan CamScanner

Pak Dengklek adalah seorang peternak bebek. Suatu hari ia mendata bebek-bebeknya dan didapatkan data berikut :

Keterangan :

- Nama : Nama bebek
- Gender : Jenis kelamin bebek
- Warna : Warna bulu bebek
- Tinggi : Tinggi badan bebek (cm)
- Lebar : Lebar badan bebek (cm)
- Berat : Berat badan bebek (kg)

6. Karena nama-nama bebeknya yang mirip satu sama lain, Pak Dengklek pusing saat mengambil data sehingga membuat banyak kesalahan. Banyak data yang kosong, duplikat, dan negatif. Anda diminta Pak Dengklek untuk menganalisis data-data bebeknya. Apa yang Anda lakukan sebelum melakukan analisis?

7. Ambillah data-data berikut. Pertimbangkan juga dengan apa yang telah Anda lakukan pada No. 5!
   a. **(H.5 2021)** Banyak bebek Pak Dengklek dengan data yang valid
   b. Ada berapa jenis warna bebek-bebek Pak Dengklek?
   c. **(H.5;P.5 2021)** Data bebek dengan nama Kwek
   d. Jenis kelamin Kwik
   e. Ukuran (volume) badan Kweak (Asumsikan seluruh panjang bebek adalah 45cm dan dimensi bebek adalah 50% dari Panjang*Lebar*Tinggi bebek)

8. Hitunglah statistik terhadap beberapa kolom berikut!
   a. Statistik bebek secara umum (deskripsi sederhana)
   b. Rata-rata volume bebek (Asumsikan seluruh panjang bebek adalah 45cm dan dimensi bebek adalah 50% dari Panjang*Lebar*Tinggi bebek)
   c. **(UAS 2021)** Rata-rata massa jenis bebek
   d. Perbandingan banyaknya bebek jantan dan betina
   e. **(UAS;H.5;P.5 2021)** Bagaimana korelasi antara tinggi badan bebek dengan berat badannya? Berapa nilai koefisien korelasinya? Jelaskan!

9. Jawablah pertanyaan-pertanyaan berikut dan buktikan dengan membuat program :
   a. **(H.5 2021)** Siapa bebek yang paling tinggi?
   b. Bebek kurus adalah bebek yang memiliki dimensi besar namun berat badan kecil. Siapa bebek yang paling kurus?
   c. Bebek gemuk adalah bebek yang memiliki dimensi kecil namun berat badan besar. Siapa bebek yang paling gemuk?
   d. Siapa bebek cokelat yang paling ringan?
   e. Siapa bebek betina yang paling besar?

10. Buatlah tabel yang memuat data-data berikut :
   a. 5 data bebek cokelat paling ringan
   b. **(P.5 2021)** Bebek jantan yang berwarna putih
   c. **(P.5 2021)** Bebek hitam yang tingginya di atas 60cm
   d. **(H.5 2021)** Bebek betina yang memiliki berat >=2kg
   e. **(H.5; P.5 2021)** 5 bebek dengan massa jenis terbesar

Dipindai dengan CamScanner

PEMBAHASAN

1. Apabila kita perhatikan, data-data N/A terletak pada kategori, dan kota. Data-data tersebut tidak bisa kita perbaiki, karena tidak diberikan referensi lain. Oleh karena itu, data N/A akan dihapus. Selanjutnya untuk data yang negatif, apabila kita perhatikan, terletak di harga tiket masuk. Kemungkinan ini adalah kesalahan input yang seharusnya positif menjadi negatif. Oleh karena itu, untuk data negatif kita akan buat menjadi positif.
   a. Untuk menghilangkan data-data N/A dapat melakukan :

```
import pandas as pd

# membaca file excel
df = pd.read_excel('Pariwisata.xlsx')
# menghapus baris yang memiliki nilai N/A
df.dropna(inplace=True)
# menampilkan data hasil
print(df)
```

   Parameter inplace=True artinya kita akan menghapus nilai N/A pada tabel df. Jika inplace=False, artinya kita akan menghapus nilai N/A dan membuat tabel salinan. Misalkan df_bersih = df.dropna(inplace=False). Dataframe df_bersih akan bersih dari nilai N/A, tetapi dataframe df masih terdapat nilai N/A.

   b. Untuk mengubah data negatif menjadi positif dapat melakukan :

```
# mengubah data negatif pada kolom price menjadi positif
df["Price"]=df["Price"].abs()
# menampilkan data hasil
print(df)
```

   Fungsi .abs() adalah untuk meng-absolut-kan (memutlakkan) nilai.

2. Berikut adalah pembahasan soal No.2 :
   a. Untuk menampilkan banyaknya tempat wisata (banyak data), kita dapat menggunakan len().

```
# Ambil banyak data
banyakdata = len(df)
# Tampilkan hasil
print(f"Banyak tempat wisata adalah : {banyakdata}")
```

   b. Untuk menampilkan 10 data pertama, kita dapat menggunakan format df[a:b] dengan a adalah awal dan b adalah akhir. Karena kita mengambil data awla, maka a adalah 0 (atau tidak perlu ditulis).

```
print(df[:10])
```

   c. Untuk menampilkan data ke-30 hingga 40, lakukan seperti pada No. 2b. Namun jangan sampai terkecoh, perlu diingat bahwa data dimulai dari 0, sehingga data ke-30 adalah indeks ke-29, dan data ke-40 adalah indeks ke-39 :

```
print(df[29:39])
```

   d. Untuk menampilkan data Gedung Sate, kita ambil dengan fungsi loc dengan parameter df['Place_Name'] == 'Gedung Sate' untuk mengambil data yang pada kolom *'Place_Name'* memiliki nilai *'Gedung Sate'* :

Dipindai dengan CamScanner

```
1. # Menampilkan data tempat wisata Gedung Sate
2. data_gedung_sate = df.loc[df['Place_Name'] == 'Gedung Sate']
3. # Menampilkan hasil
4. print(data_gedung_sate)
```

e. Kita telah mengambil data gedung sate. Kita dapat menggunakan variabel tersebut untuk mengambil data harga tiket masuk dengan cara berikut :

```
1. # Cari harga tiket masuk
2. htm_gedung_sate = data_gedung_sate['Price'].values[0]
3. # Menampilkan hasil
4. print(f"Harga tiket Gedung Sate : Rp{htm_gedung_sate}")
```

3. Berikut pembahasan untuk soal No.3 :

a. Untuk mencari rata-rata rating, gunakan fungsi .mean() pada kolom Rating :

```
1. ratarata = df['Rating'].mean()
2. print(f'Rata-rata penilaian tempat wisata adalah {rata-rata}')
```

b. Untuk mencari standar deviasi harga, gunakan fungsi .std() pada kolom Price :

```
1. std_harga = df['Price'].std()
2. print(f'Standar deviasi harga tiket masuk tempat wisata adalah
{std_harga}')
```

c. Untuk mencari jumlah tempat wisata di kota Bandung, kita ambil data dengan fungsi loc[] dan dengan parameter City = "Bandung". Lalu gunakan fungsi len() untuk mengambil banyak data.

```
1. wisata_bandung = df.loc[df['City'] == 'Bandung']
2. count_bandung = len(wisata_bandung)
3. print(f'Jumlah tempat wisata di Bandung adalah {count_bandung}')
```

d. Pertama-tama kita harus mengelompokkan data berdasarkan kota dengan fungsi .groupby(). Setelah itu kita gunakan fungsi .count() untuk menghitung banyak data berdasarkan pengelompokan tersebut (frekuensi).

```
1. wisata_per_kota = df.groupby('City').size()
2. print('Jumlah tempat wisata di tiap kota:')
3. print(wisata_per_kota)
4.
```

e. Untuk mencari nilai korelasi, kita gunakan fungsi .corr() sebagai berikut :

```
1. # Ambil data tempat wisata yang tidak gratis
2. berbayar = df.loc[df['Price'] != 0]
3. # Ambil nilai korelasi antara harga dengan rating
4. korelasi = berbayar['Price'].corr(berbayar['Rating'])
5. # Menampilkan hasil
6. print(f'Korelasi antara harga tiket masuk dengan rating tempat wisata
adalah {korelasi}')
```

Hasil yang diperoleh adalah 0.01, yang artinya harga tiket masuk dengan rating tempat wisata tidak memiliki korelasi.

Dipindai dengan CamScanner

4. Berikut pembahasan untuk soal No. 4 :

a. Untuk memperoleh kota dari Situ Cileunca, kita harus mengambil datanya terlebih dahulu. Barulah kita ambil kota dari Situ Cileunca tersebut.

```
1. # Ambil data Situ Cileunca
2. situcileunca = df[df['Place_Name']=='Situ Cileunca']
3. # Ambil nama kota dari Situ Cileunca
4. kota_situcileunca = situcileunca['City'].values[0]
5. # Tampilkan hasil
6. print(f'Situ Cileunca terletak di {kota_situcileunca}.')
```

b. Untuk mendapatkan tempat wisata termahal, pertama kita cari terlebih dahulu berapa harga tiket masuk termahal. Lalu cari data yang harga tiketnya sama dengan harga tiket masuk termahal yang telah kita cari. Setelah itu, ambil nama tempat wisata tersebut.

```
1. # Cari nilai maksimum dari Price (HTM termahal)
2. htm_termahal = df['Price'].max()
3. # Ambil data tempat wisata dengan HTM termahal
4. wisata_mahal = df[df['Price']==htm_termahal]
5. # Ambil nama tempat wisata
6. nama_wisata_mahal = wisata_mahal['Place_Name'].values[0]
7. # Tampilkan hasil
8. print(f'Tempat wisata termahal adalah {nama_wisata_mahal}.')
```

c. Pertama-tama kita harus memfilter data agar selanjutnya yang kita analisis adalah tempat wisata di Yogyakarta. Setelah itu, cari nilai maksimum dari rating. Barulah kita cari tempat wisata dengan rating maksimum sesuai yang kita dapatkan.

```
1. # Filter untuk tempat wisata di Yogyakarta
2. wisata_jogja = df[df['City']=='Yogyakarta']
3. # Ambil rating terbaik tempat wisata di Yogyakarta
4. ratingterbaik_jogja = wisata_jogja['Rating'].max()
5. # Ambil data tempat wisata di Yogyakarta dengan rating terbaik
6. best_rate_jogja =
   wisata_jogja[wisata_jogja['Rating']==ratingterbaik_jogja]
7. # Tampilkan hasil
8. print(best_rate_jogja)
```

d. Pertama, ambil data untuk Masjid Salman ITB. Setelah itu, ambil harga tiket masuknya. Buatlah percabangan untuk menentukan apakah Masjid Salman ITB gratis atau berbayar. Bila berbayar, maka sebutkan harganya.

```
1. # Ambil data Masjid Salman ITB
2. salmanitb = df[df['Place_Name']=='Masjid Salman ITB']
3. # Ambil harga tiket masuk
4. htm_salman = salmanitb['Price'].values[0]
5. # Cek apakah Masjid Salman ITB berbayar/gratis
6. if (htm_salman==0):
7.     print("Masjid Salman ITB gratis.")
8. else:
9.     print(f"HTM dari Masjid Salman ITB adalah {htm_salman}.")
```


Dipindai dengan CamScanner

e. Ada beberapa tahap untuk mendapatkan informasi tersebut. Pertama, kita harus memfilter agar data yang selanjutnya kita naalisis adalah tempat wisata yang ada di Bandung. Kedua, cari nilai maksimum dari kolom Price (HTM termahal). Ketiga, ambil data tempat wisata dengan nilai maksimum yang telah diperoleh. Keempat, ambil kategori tempat wisata tersebut. Setelah itu, baru tuliskan hasilnya.

```
1. # Ambil data tempat wisata di Bandung
2. wisata_bandung = df.loc[df['City'] == 'Bandung']
3. # Ambil data HTM termahal
4. htm_termahal = wisata_bandung['Price'].max()
5. # Ambil data tempat wisata dengan HTM termahal
6. wisata_bandung_mahal =
   wisata_bandung[wisata_bandung['Price']==htm_termahal]
7. # Ambil kategori tempat wisata tersebut
8. kategori_wisata_bandung_mahal =
   wisata_bandung_mahal['Category'].values[0]
9. # Tampilkan hasil
10. print(f"Kategori wisata di Bandung termahal adalah
{kategori_wisata_bandung_mahal}.")
```

5. Berikut pembahasan untuk soal No. 5 :

a. Untuk mecari 5 tempat wisata termurah, kita harus kecualikan tempat wisata yang gratis terlebih dahulu. Hal ini telah kita dapatkan pada nomor 3e. Lalu urutkan data secara ascending (terkecil ke terbesar), dan ambil 5 data pertama.

```
1. # Urutkan data dari yang terkecil (ascending)
2. terurut_asc = berbayar.sort_values(['Price'], ascending=1)
3. # Tampilkan 5 data pertama
4. print(terurut_asc[:5])
```

b. Untuk memperoleh 10 tempat wisata termahal, kita ulangi langkah No. 5a namun dengan urutan descending. Caranya adalah mengubah parameter ascending menjadi 0, lalu tampilkan 10 data pertama.

```
1. # Urutkan data dari yang terbesar (descending)
2. terurut_des = berbayar.sort_values(['Price'], ascending=0)
3. # Tampilkan 5 data pertama
4. print(terurut_des[:10])
```

c. Gunakan boolean df['Rating']>45 untuk mengambil data yang memiliki rating lebih dari 45 :

```
1. # Ambil data yang memiliki rating di atas 45
2. Rating45 = df[df['Rating'] > 45]
3. # Tampilkan hasil
4. print(Rating45)
```

d. Untuk mengambil data tempat wisata cagar alam di Yogyakarta, maka kita gunakan 2 boolean dan gabungkan dengan operator and (&). Dua boolean yang dimaksud adalah dan :

```
1. # Ambil data tempat wisata cagar alam di Yogyakarta
```


Dipindai dengan CamScanner

```
2. cagaralam_jogja = df[(df['City'] == 'Yogyakarta') & (df['Category']
   == 'Cagar Alam')]
3. # Tampilkan hasil
4. print(cagaralam_jogja)
```

e. Untuk mengambil data jumlah tempat wisata gratis di masing-masing kota, pertama kita harus mengambil data tempat wisata yang gratis terlebih dahulu. Barulah kita gabungkan berdasarkan kota dengan fungsi .groupby() dan hitung frekuensinya menggunakan .count() :

```
1. # Ambil tempat wisata gratis
2. gratis = df[df['Price']==0]
3. # Kelompokkan berdasarkan kota
4. gratis_per_kota = gratis.groupby('City').size()
5. # Tampilkan hasil
6. print(gratis_per_kota)
```

6. Soal ini kurang-lebih sama dengan No.1. Sebelum melakukan analisis, ada beberapa langkah yang perlu dilakukan.

a. Menghapus data duplikat

Data duplikat perlu kita hapus, karena dapat memengaruhi hasil analisis. Caranya adalah sebagai berikut :

```
1. import pandas as pd
2.
3. # Membaca file csv
4. data_bebek = pd.read_csv("data_bebek.csv")
5. # Menghapus data duplikat
6. data_bebek.drop_duplicates(inplace=True)
```

b. Menghapus data kosong

Data kosong juga perlu kita hapus. Caranya adalah sebagai berikut :

```
1. # Menghapus data kosong
2. data_bebek.dropna(inplace=True)
```

c. Memeriksa data negatif

Jika kita analisis, data negatif akan terlihat normal bila dipositifkan. Oleh karena itu, data-data yang negatif akan dipositifkan dengan cara :

```
1. # Mengubah data negatif menjadi data positif
2. data_bebek['Tinggi'] = data_bebek['Tinggi'].abs()
3. data_bebek['Lebar'] = data_bebek['Lebar'].abs()
4. data_bebek['Berat'] = data_bebek['Berat'].abs()
```

7. Berikut pembahasan untuk soal No. 7 :

a. Untuk memperoleh banyak data, gunakan fungsi len() :

```
1. # Ambil banyak data
2. banyak_data_valid = len(data_bebek)
3. # Tuliskan hasil
4. print(f"Banyak bebek Pak Dengklek adalah: {banyak_data_valid}")
```

b. Gunakan fungsi .nunique() untuk mendapatkan jumlah kategori unik dalam suatu kolom :

Dipindai dengan CamScanner

```
1. # Ambil jumlah kategori unik dalam kolom Warna
2. jenis_warna = data_bebek['Warna'].nunique()
3. # Tuliskan hasil :
4. print(f"Ada {jenis_warna} jenis warna bebek.")
```

c. Untuk memperoleh data bebek dengan nama Kwek, caranya adalah :

```
1. # Ambil data bebek dengan nama Kwek
2. data_kwek = data_bebek[data_bebek['Nama'] == 'Kwek']
3. # Tuliskan hasil
4. print(data_kwek)
```

d. Kita telah mengambil data Kwek. Selanjutnya, kita tinggal mengambil jenis kelamin Kwek dengan data tersebut :

```
1. # Ambil nilai jenis kelamin Kwek
2. jenis_kelamin_kwek = data_kwek['Gender'].values[0]
3. # Tuliskan hasil
4. print(f"Jenis kelamin Kwek adalah {jenis_kelamin_kwek}.")
```

e. Ada beberapa tahap yang perlu kita lakukan. Pertama, tulis asumsi bahwa panjang seluruh bebek adalah 45cm. Kedua, ambil data Kweak. Ketiga, ambil tinggi dan lebar Kweak. Terakhir, hitung ukuran badan bebek sebagai panjang*lebar*tinggi/2 dan tuliskan hasilnya :

```
1. # Tuliskan asumsi panjang bebek
2. panjang_bebek = 45
3. # Ambil data Kweak
4. data_kweak = data_bebek[data_bebek['Nama'] == 'Kweak']
5. # Ambil lebar dan tinggi Kweak
6. lebar_kweak = data_kweak['Lebar'].values[0]
7. tinggi_kweak = data_kweak['Tinggi'].values[0]
8. # Hitung ukuran badan kweak dan tuliskan
9. ukuran_kweak = 0.5 * panjang_bebek * lebar_kweak * tinggi_kweak
10. print(f"Ukuran (volume) badan Kweak: {ukuran_kweak}cm^3")
```

8. Berikut adalah pembahasan untuk soal No. 8 :

a. Untuk menampilkan statistik secara umum, gunakan fungsi .describe() :

```
1. print(data_bebek.describe())
```

b. Untuk membuat rata-rata volume bebek, kita langsung lakukan operasi pada dataframe. Operasi tersebut akan menghasilkan dataframe baru sebagai hasil dari operasi.

```
1. # Tuliskan asumsi
2. panjang_bebek = 45
3. # Lakukan operasi terhadap dataframe
4. volume_bebek = 0.5 * panjang_bebek * data_bebek['Lebar'] *
   data_bebek['Tinggi']
5. # Hitung rata-rata volume bebek
6. rata2_volume_bebek = volume_bebek.mean()
7. # Tuliskan hasil
8. print(f"Rata-rata volume bebek: {rata2_volume_bebek}cm^3")
```


Dipindai dengan CamScanner

c. Kita telah memperoleh data volume bebek pada sub-soal sebelumnya. Oleh karena itu, kita tinggal membagi berat dengan volume dan hitung rata-ratanya. Berat dikalikan 1000 untuk dikonversi menjadi gram.

```
1. # Hitung massa jenis bebek
2. rho_bebek = data_bebek['Berat']*1000 / volume_bebek
3. # Hitung rata-rata massa jenis bebek
4. rata2_rho_bebek = rho_bebek.mean()
5. # Tuliskan hasil
6. print(f"Rata-rata massa jenis bebek: {rata2_rho_bebek}g/cm^3")
```

d. Pertama, kita harus cari jumlah bebek jantan dan betina. Lalu gunakan perbandingan dengan membagi jantan/betina. Nilai tersebut akan merepresentasikan jumlah bebek jantan adalah sekian kali dari bebek betina. Implementasi dalam python adalah sebagai berikut :

```
1. # Cari banyak bebek jantan dan betina
2. banyak_jantan = len(data_bebek[data_bebek['Gender'] == 'Jantan'])
3. banyak_betina = len(data_bebek[data_bebek['Gender'] == 'Betina'])
4. # Bandingkan banyak bebek jantan dan betina dan tuliskan hasilnya
5. perbandingan = banyak_jantan / banyak_betina
6. print(f"Jumlah bebek jantan adalah {perbandingan}x dari bebek
   betina.")
```

e. Cari nilai korelasi dengan fungsi .corr() :

```
1. # Cari nilai korelasi dan tuliskan hasilnya
2. korelasi = data_bebek['Tinggi'].corr(data_bebek['Berat'])
3. print(f"Nilai korelasi : {korelasi}")
```

Nilai koefisien korelasi yang diperoleh adalah 0.742546. Artinya, hubungan antara tinggi bebek dengan berat bebek adalah berbanding lurus. Semakin tinggi bebek, maka semakin berat.

9. Berikut adalah pembahasan untuk No. 9 :

a. Untuk mencari tahu siapa bebek yang paling tinggi, ada beberapa tahapan. Pertama, cari tinggi bebek yang paling tinggi. Kedua, cari data bebek yang memiliki tinggi tersebut. Ketiga, ambil nama dari data bebek yang telah diperoleh. Terakhir, tuliskan hasilnya.

```
1. # Cari tinggi bebek yang paling tinggi
2. tinggi_max = data_bebek['Tinggi'].max()
3. # Cari data bebek yang memiliki tinggi tersebut
4. bebek_tinggi = data_bebek[data_bebek['Tinggi'] == tinggi_max]
5. # Ambil nama bebek dari data tersebut
6. nama_bebek_tinggi = bebek_tinggi['Nama'].values[0]
7. # Tuliskan hasilnya
8. print(f"Bebek tertinggi adalah {nama_bebek_tinggi} dengan tinggi
   {tinggi_max} cm.")
```

b. Kita telah membuat dataframe massa jenis pada soal 8c. Bebek kurus adalah bebek dengan volume besar dan berat kecil. Dengan begitu, kita tinggal mencari bebek dengan berat/volume (massa jenis) terkecil. Gabungkan data

Dipindai dengan CamScanner

massa jenis dengan dataframe utama. Setelah itu, kita cari tahu siapa bebek tersebut. Perhatikan kode berikut :

```
# Cari bebek terkurus
terkurus = rho_bebek.min()
# Gabungkan kolom rho_bebek dengan dataframe utama (data_bebek)
data_bebek = pd.concat([data_bebek, rho_bebek.rename("Rho")],
axis=1)
# Ambil data bebek dengan massa jenis terkecil (terkurus)
bebek_kurus = data_bebek[data_bebek["Rho"]==terkurus]
# Ambil nama bebek dari data tersebut dan tuliskan namanya
nama_bebek_kurus = bebek_kurus["Nama"].values[0]
print(f"Bebek terkurus adalah {nama_bebek_kurus}")
```

c. Kita telah menggabungkan data rho dengan dataframe utama. Untuk mencari bebek tergemuk, maka kita cari bebek dengan massa jenis terbesar.

```
# Cari bebek terkurus
tergemuk = rho_bebek.max()
# Ambil data bebek dengan massa jenis terbesar (tergemuk)
bebek_gemuk = data_bebek[data_bebek["Rho"]==tergemuk]
# Ambil nama bebek dari data tersebut dan tuliskan namanya
nama_bebek_gemuk = bebek_gemuk["Nama"].values[0]
print(f"Bebek tergemuk adalah {nama_bebek_gemuk}")
```

d. Pertama, kita filterkan untuk bebek berwarna cokelat. Lalu cari berat bebek paling ringan. Ambil data untuk bebek yang memiliki berat tersebut.

```
# Filter untuk warna bebek cokelat
bebek_cokelat = data_bebek[data_bebek["Warna"]=="Cokelat"]
# Cari berat bebek yang paling ringan dari bebek cokelat
bebek_cokelat_ringan = bebek_cokelat["Berat"].min()
# Cari data bebek dengan berat tersebut
data_cokelat_ringan =
bebek_cokelat[bebek_cokelat["Berat"]==bebek_cokelat_ringan]
# Ambil nama dari bebek tersebut
nama_cokelat_ringan = data_cokelat_ringan["Nama"].values[0]
# Tuliskan nama bebek
print(f"Bebek cokelat yang paling ringan adalah :
{nama_cokelat_ringan}")
```

e. Pertama, Gabungkan data volume yang telah dibuat pada soal 8b lalu filterkan untuk bebek betina. Cari volume bebek betina terbesar. Ambil data untuk bebek betina yang memiliki volume tersebut dan ambil namanya :

```
# Gabungkan kolom volume_bebek dengan dataframe utama
data_bebek = pd.concat([data_bebek, volume_bebek.rename("Volume")],
axis=1)
# Filter untuk warna bebek betina
bebek_betina = data_bebek[data_bebek["Gender"]=="Betina"]
# Cari volume bebek yang paling besar dari bebek betina
bebek_betina_besar = bebek_betina["Volume"].max()
# Cari data bebek dengan volume tersebut
```


Dipindai dengan CamScanner

```
8. data_betina_besar =
bebek_betina[bebek_betina["Volume"]==bebek_betina_besar]
9. # Ambil nama dari bebek tersebut
10. nama_betina_ringan = data_betina_besar["Nama"].values[0]
11. # Tuliskan nama bebek
12. print(f"Bebek betina yang paling besar adalah :
    {nama_betina_ringan}")
```

10. Berikut adalah pembahasan untuk soal No. 10 :

a. Ada beberapa tahap untuk mendapatkan 5 data bebek cokelat paling ringan. Pertama, filter data untuk bebek berwarna cokelat. Urutkan data berdasarkan beratnya secara ascending. Ambil 5 data pertama.

```
1. # Filter data berwarna cokelat
2. data_cokelat = data_bebek[data_bebek['Warna'] == 'Cokelat']
3. # Urutkan data berdasarkan berat dari yang terkecil
4. data_cokelat_ringan = data_cokelat.sort_values(['Berat'],
   ascending=1)
5. # tampilkan tabel
6. print(data_cokelat_ringan[:5])
```

b. Untuk memperoleh bebek jantan berwarna putih, terdapat dua cara. Gunakan penggabungan boolean atau filter satu per satu. Berikut adalah cara filter satu per satu :

```
1. #  Filter data untuk bebek berwarna putih
2. data_putih = data_bebek[data_bebek['Warna'] == 'Putih']
3. # Filter data untuk bebek jantan berwarna putih
4. data_putih_jantan = data_putih[data_putih['Gender'] == 'Jantan']
5. # Tampilkan hasil
6. print(data_putih_jantan)
```

c. Untuk memperoleh bebek hitam yang tingginya di atas 60cm, terdapat dua cara. Gunakan penggabungan boolean atau filter satu per satu. Berikut adalah cara penggabungan boolean :

```
1. # filter data berwarna hitam dan tinggi di atas 60cm
2. data_hitam_tinggi = data_bebek[(data_bebek['Warna'] == 'Hitam') &
   (data_bebek['Tinggi'] > 60)]
3. # Tampilkan hasil
4. print(data_hitam_tinggi)
```

d. Untuk memperoleh bebek betina yang memiliki berat >=2kg, terdapat dua cara. Gunakan penggabungan boolean atau filter satu per satu. Berikut adalah cara penggabungan boolean :

```
1. # Filter data bebek betina dengan berat >= 2kg
2. betina_2kg = data_bebek[(data_bebek['Gender'] == 'betina') &
   (data_bebek['Berat'] >= 2)]
3. # Tampilkan hasil
4. print(betina_2kg)
```


Dipindai dengan CamScanner

e. Kita telah menggabungkan data massa jenis pada dataframe utama (data_bebek) pada nomor 9b. Untuk memperoleh 5 bebek dengan massa jenis terbesar, urutkan data dari yang terbesar (descending), lalu ambil 5 data pertama.

```
1. # Urutkan data massa jenis secara descending
2. data_mj_terbesar = data_bebek.sort_values('Rho', ascending=False)
3. # Tampilkan 5 data pertama
4. print(data_mj_terbesar[:5])
```


Dipindai dengan CamScanner

BAB X: Visualisasi Data

VISUALISASI DATA

Visualisasi data adalah merepresentasikan data dalam bentuk visual. Tujuannya adalah untuk analisis data (exploratory) dan komunikasi (explanatory). Untuk melakukan visualisasi, tahap pertama adalah memuat data dan library, misalnya :

```
1. import pandas as pd
2. import matplotlib.pyplot as plt
3. df = pd.read_csv("namafile.csv")
```

JENIS-JENIS VISUALISASI DAN CARA MEMBUATNYA

Untuk membuat visualisasi, pilihlah jenis visualisasi yang cocok untuk menggambarkan data. Berikut ini jenis-jenis visualisasi dan cara menginisiasinya :

COMPARING CATEGORIES

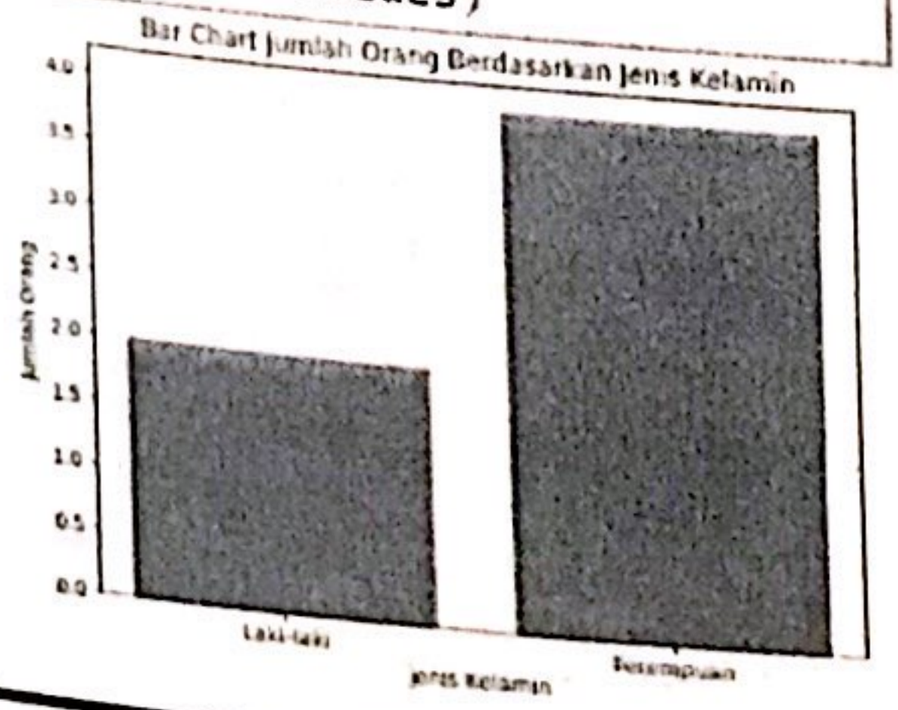
Yaitu untuk membandingkan nilai-nilai dari suatu variabel yang bersifat kategorikal.

- **Bar Chart**

```
1. plt.bar(index,values)
```

Contoh :

```
1. # Bar chart jumlah orang
   berdasarkan jenis kelamin
2. gender =
   df.groupby('Gender').size()
3. plt.bar(gender.index,
   gender.values)
```

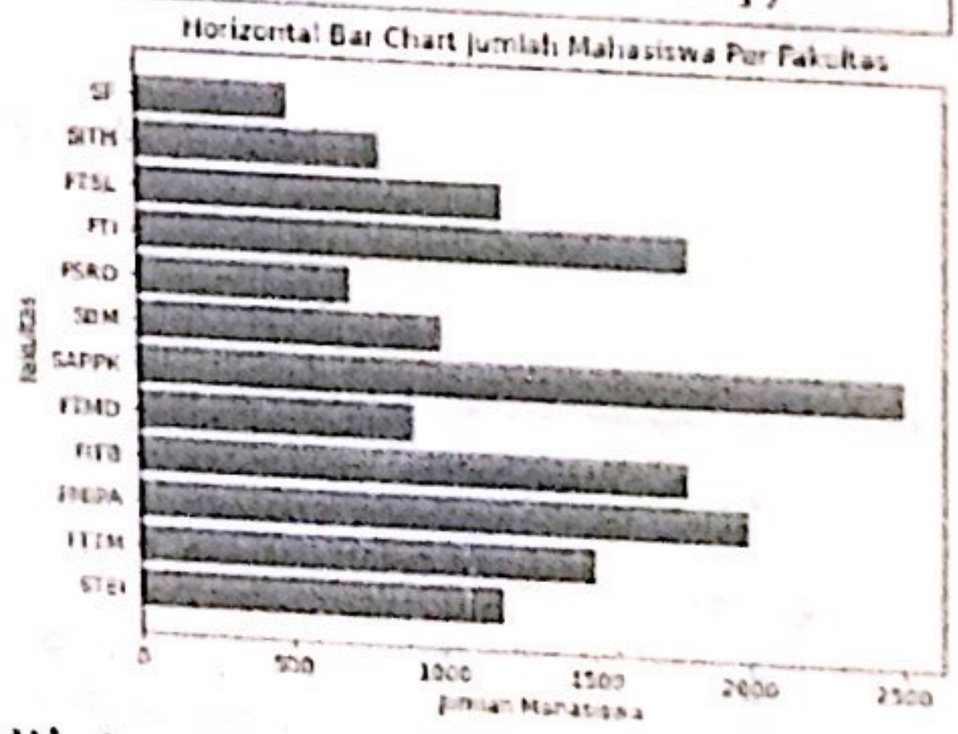


- **Horizontal Bar Chart**

```
1. plt.barh(index,values)
```

Contoh :

```
1. # Bar chart jumlah
   mahasiswa per fakultas
2. plt.barh(df['Fakultas'],
   df['Jumlah_Mahasiswa'])
```

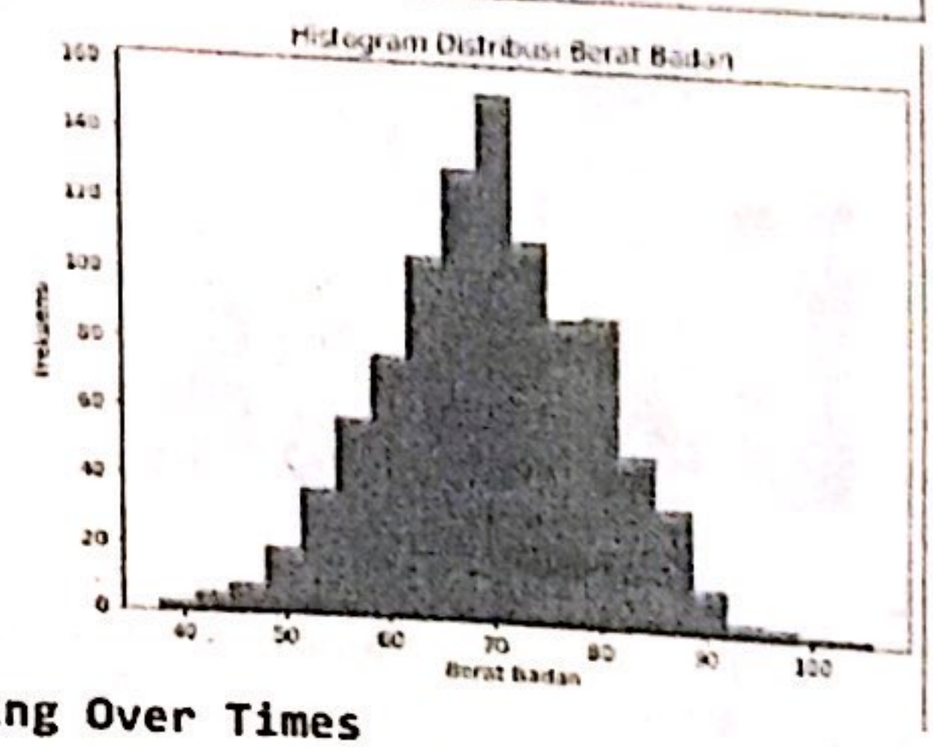


- **Histogram**

```
1. plt.hist(data, bins=n)
```

Contoh :

```
1. # Distribusi berat badan
dengan bins 10
2. Plt.hist(df['Berat'],
bins=10)
```



Showing Over Times

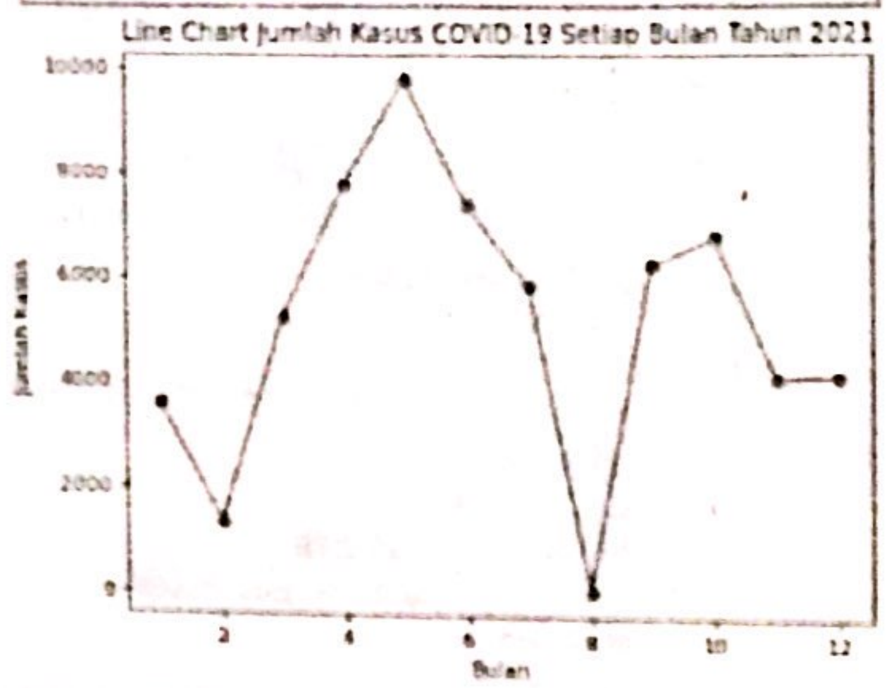
Yaitu untuk menampilkan data dalam periode waktu tertentu.

- **Line Chart**

```
1. plt.plot(x,y)
```

Contoh :

```
1. # Jumlah kasus COVID-19
   tiap bulan tahun 2021
2. plt.plot(df['Bulan'],
   df['Kasus'])
```

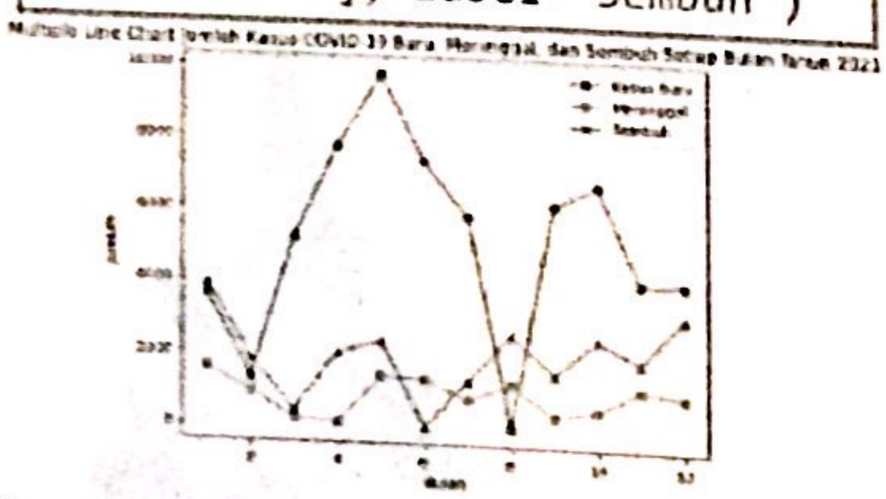


- **Multiple Line Chart**

```
1. plt.plot(x,y)
```

Contoh :

```
1. # Jumlah kasus COVID-19
baru, meninggal, dan sembuh
per bulan tahun 2021
2. plt.plot(df['Bulan'],
df['Kasus'], label='Kasus
baru')
3. plt.plot(df['Bulan'],
df['Meninggal'],
label='Meninggal')
4. plt.plot(df['Bulan'],
df['Sembuh'], label='Sembuh')
```

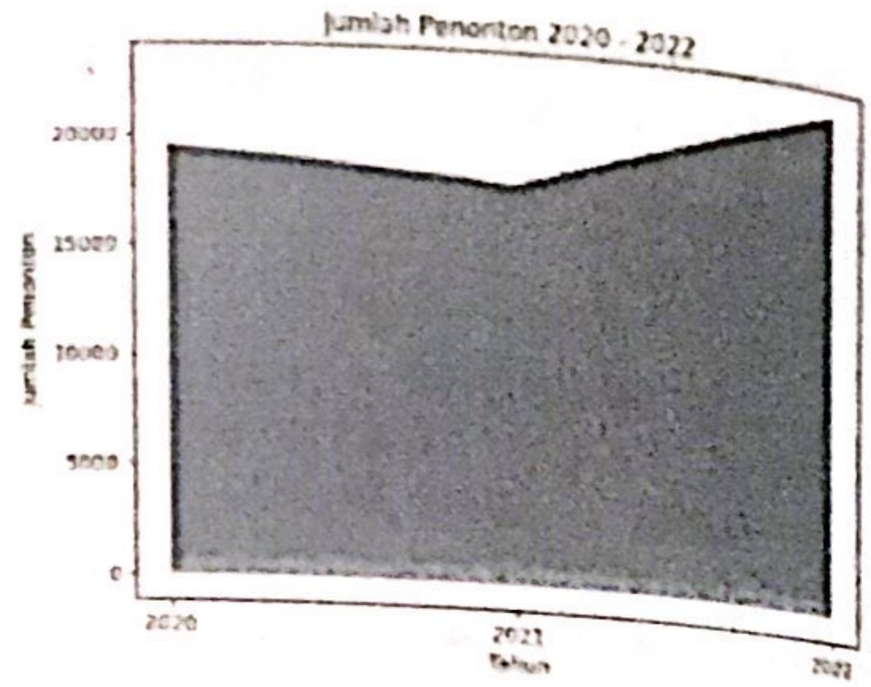


- **Area Chart**

```
1. plt.fill_between(x,y)
```

Contoh :

```
1. # Jumlah penonton dari
tahun ke tahun
2.
plt.fill_between(df['Tahun'],
df['Penonton'])
```

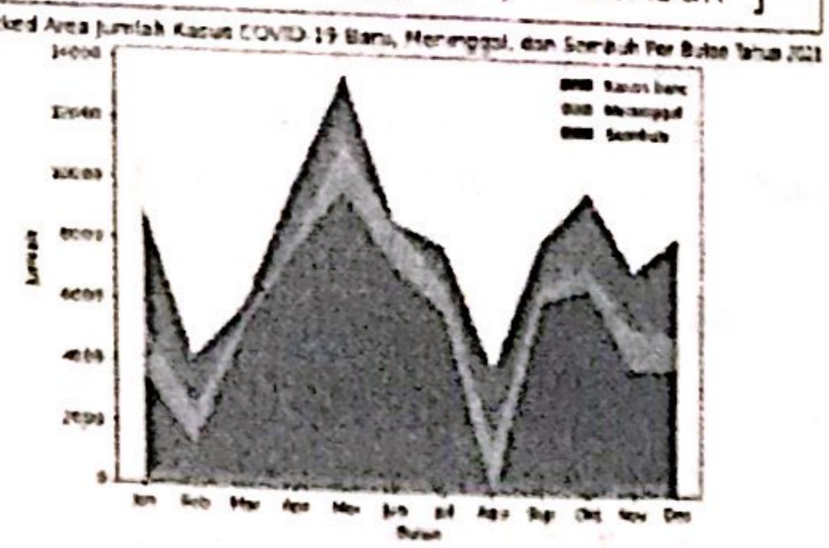


- **Stacked Area Chart**

```
1. plt.stackplot(x, y1, y2,
..., labels=['Area1', 'Area2',
...])
```

Contoh :

```
1. # Jumlah kasus COVID-19
baru, meninggal, dan sembuh
per bulan tahun 2021
2. plt.stackplot(df['Bulan'],
df['Kasus'], df['Meninggal'],
df['Sembuh'], labels=['Kasus
baru', 'Meninggal', 'Sembuh']
```



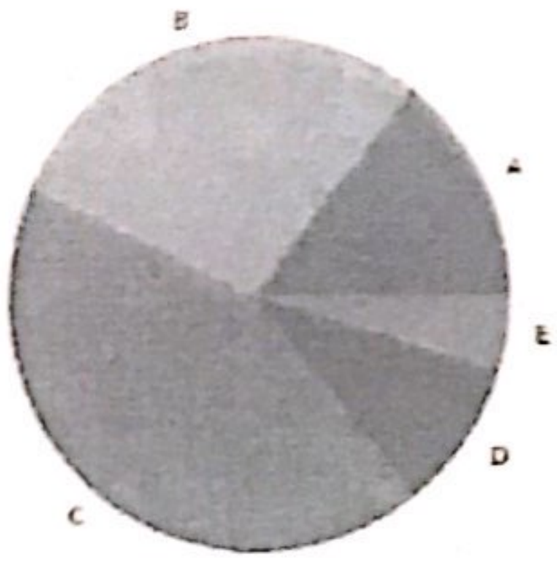
Hierarchy and Whole-part Relationship
Hierarki adalah menampilkan urutan data. Whole-part relationship adalah menampilkan proporsi bagian-bagian suatu variabel dari keseluruhannya.

- **Pie chart**

```
1. plt.pie(ukuran,
labels=label)
```

Contoh :

```
1. plt.pie(df['Jumlah_Kamar'],
labels=df['Jenis_Kamar'])
```

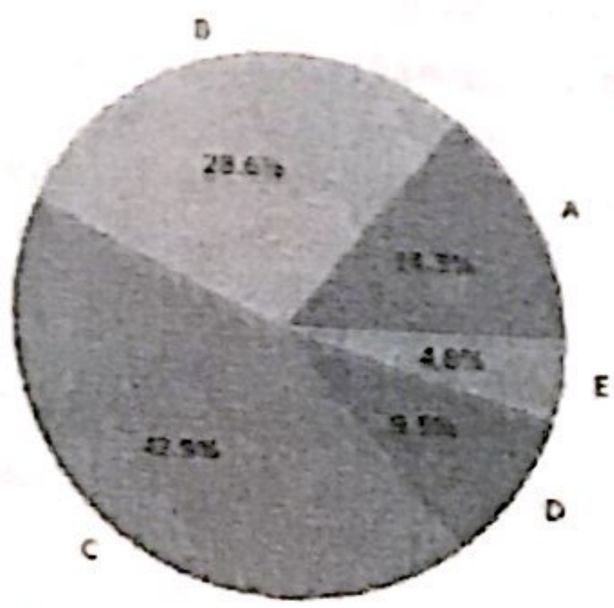


- **Pie chart with percent**

```
1. plt.pie(ukuran,
labels=label,
autopct='%1.1f%%')
```

Contoh :

```
1. plt.pie(df['Jumlah_Kamar'],
labels=df['Jenis_Kamar'],
autopct='%1.1f%%')
```

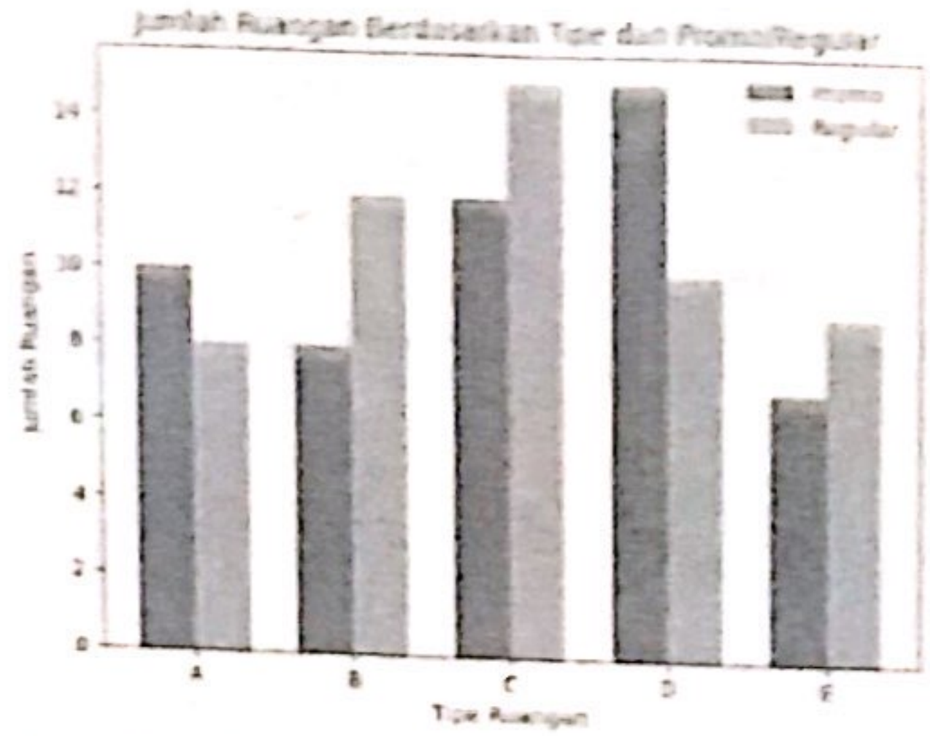


- **Multiple Bar Chart**

```
1. df.plot(title="Judul",
x="KolomX", y=["KolomY1",
"KolomY2", ...])
```

Contoh :

```
1. df.plot(title="Jenis Kamar
terhadap
Promo/Regular",kind="bar",
x="Jenis_Kamar",
y=["Jumlah_Promo",
"Jumlah_Regular"])
```

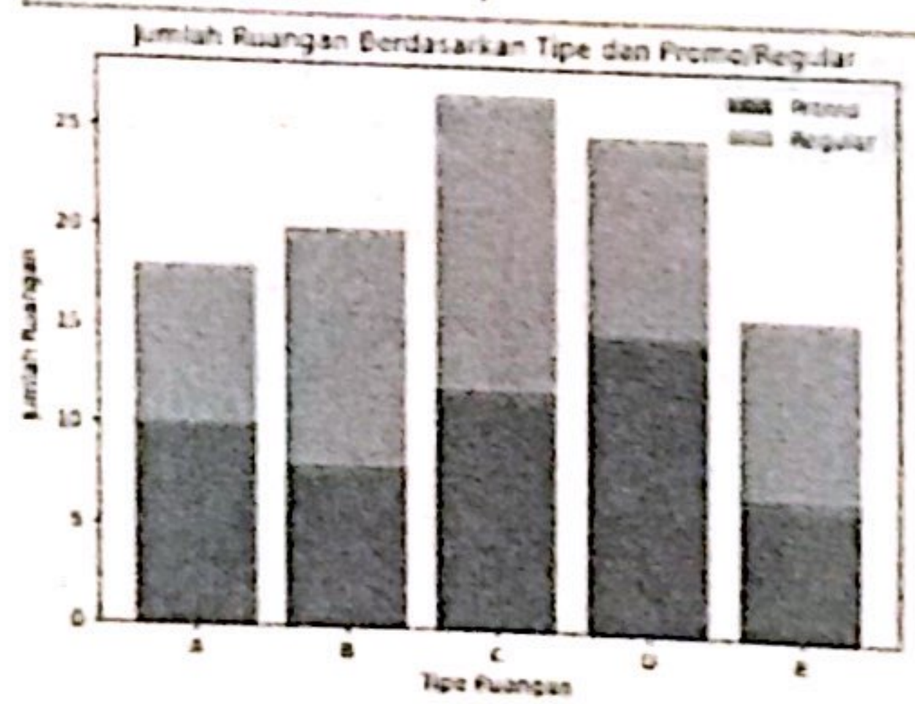


- **Stacked Bar Chart**

```
1. fig, ax = plt.subplots()
2. ax.bar(x, y,
label="label1")
3. ax.bar(x, y,
label="label2")
...
```

Contoh :

```
1. fig, ax = plt.subplots()
2. ax.bar(df["Jenis_Kamar"],
df["Jumlah_Promo"],
label="Promo")
3. ax.bar(df["Jenis_Kamar"],
df["Jumlah_Regular"],
label="Regular")
```



Plotting Relationshop

Yaitu menunjukkan korelasi antar data dengan 2 variabel atau lebih.

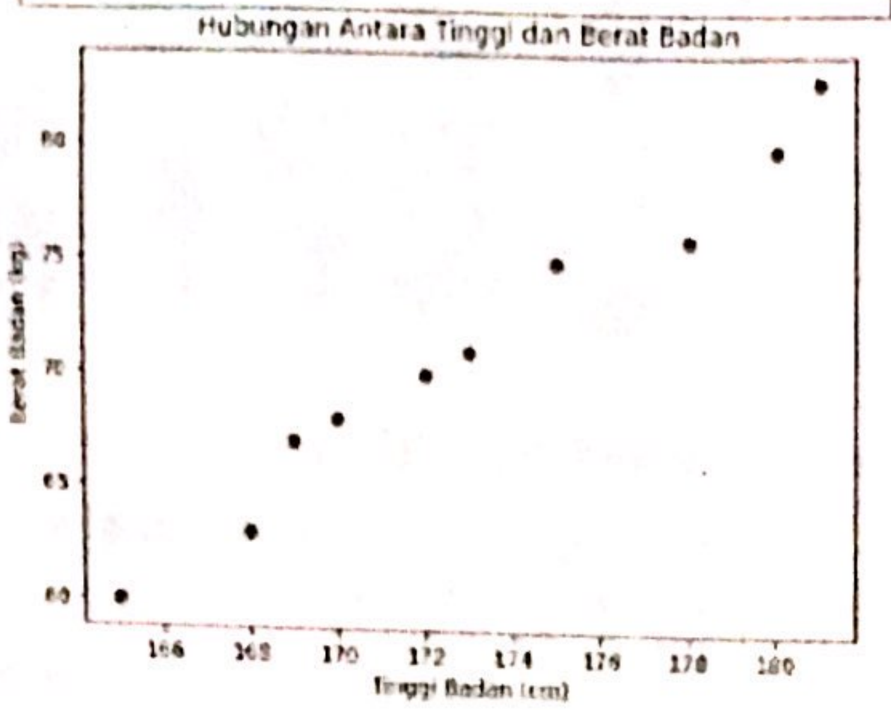
- **Scatter plot**

```
1. plt.scatter(x,y)
```

Dipindai dengan CamScanner

Contoh :

```
1. # Hubungan antara tinggi
dan berat badan
2. plt.scatter(df["Tinggi"],
df["Berat"])
```

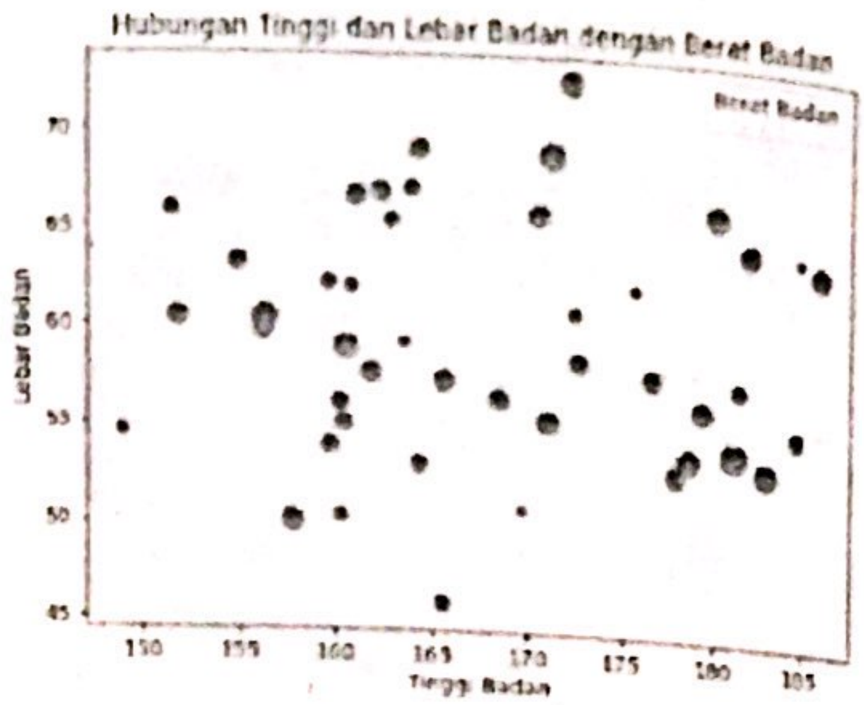


- **Bubble plot**

```
1. plt.scatter(x,y,s=sizes)
```

Contoh :

```
1. # Hubungan tinggi dan lebar
badan dengan size
menggambarkan berat badan :
2. plt.scatter(df["Tinggi"],
df["Lebar"], s=df["Berat"])
```



Setelah memilih dan menginisiasi visualisasi, (kecuali Multiple bar chart) langkah selanjutnya adalah memberi judul grafik, label sumbu x dan sumbu y dengan cara berikut :

```
1. # Memberi judul dan label pada
sumbu x dan y
2. plt.title('Judul Grafik')
3. plt.xlabel('Sumbu X')
4. plt.ylabel('Sumbu Y')
```

Langkah terakhir adalah tampilkan grafik yang telah diinisiasi dengan cara :

```
1. # menampilkan grafik
2. plt.show()
```

Versi Berwarna :

SOAL-SOAL

Untuk 5 soal berikutnya, silahkan scan QR Code berikut untuk mengunduh data :

Keterangan :

- Place_Name : Nama tempat wisata
- Category : Kategori tempat wisata
- City : Lokasi kota tempat wisata
- Price : Harga tiket masuk (Jika 0, berarti gratis)
- Rating : Penilaian dari pengunjung, range 0 - 50

1. **(H.5;P.5 2021)** Buatlah histogram distribusi harga tiket dengan bins 10
2. Buatlah scatter plot antara harga tiket dengan rating
3. **(H.5 2021)** Buatlah grafik untuk menggambarkan komposisi dalam persentase mengenai berapa banyak kategori tempat wisata di Indonesia. Pilih satu alternatif jenis grafik yang tepat untuk menggambarkannya. Jelaskan mengapa Anda memilih jenis grafik tersebut!
4. **(UAS 2021)** Buatlah grafik untuk menggambarkan banyaknya tempat wisata di tiap kota. Pilih satu alternatif jenis grafik yang tepat untuk menggambarkannya. Jelaskan mengapa Anda memilih jenis grafik tersebut!
5. Buatlah grafik untuk menggambarkan distribusi rating tempat wisata di Yogyakarta. Pilih satu alternatif jenis grafik yang tepat untuk menggambarkannya. Jelaskan!

Dipindai dengan CamScanner

Untuk 5 soal berikutnya, silahkan scan QR Code berikut untuk mengunduh data :

Keterangan :

- Nama : Nama bebek
- Gender : Jenis kelamin bebek
- Warna : Warna bulu bebek
- Tinggi : Tinggi badan bebek (cm)
- Lebar : Lebar badan bebek (cm)
- Berat : Berat badan bebek (kg)

6. Buatlah histogram distribusi berat badan bebek jantan dengan bins 15
7. **(P.5 2021)** Buatlah Pie chart persentase jenis kelamin bebek yang memiliki berat badan di bawah 2kg
8. **(P.5 2021)** Buatlah scatter plot dengan x sebagai tinggi badan dan y sebagai berat bebek
9. Buatlah grafik yang menggambarkan tinggi dan lebar badan 10 bebek terberat. Pilih satu alternatif jenis grafik yang tepat untuk menggambarkannya. Jelaskan mengapa Anda memilih jenis grafik tersebut!
10. **(P.5 2021)** Buatlah grafik yang menggambarkan jenis kelamin bebek terhadap warnanya. Pilih satu alternatif jenis grafik yang tepat untuk menggambarkannya. Jelaskan mengapa Anda memilih jenis grafik tersebut!

Dipindai dengan CamScanner

PEMBAHASAN

1. Untuk membuat grafik, pertama kita harus muat terlebih dahulu file yang akan digunakan dan lakukan data cleaning seperti yang dilakukan pada bab 8.

```
import pandas as pd
# Membaca file excel
df = pd.read_excel('Pariwisata.xlsx')
# Menghapus baris yang memiliki nilai N/A
df.dropna(inplace=True)
# Mengubah data negatif pada kolom price menjadi positif
df["Price"]=df["Price"].abs()
```

Barulah kita mulai membuat grafik. Untuk membuat histogram, maka gunakan plt.hist(kolom,bins=jumlah_batang), dengan kolom adalah dataframe kolom yang ingin dibuat histogram, dan jumlah_batang adalah jumlah batang pada histogram. Jangan lupa untuk mengimport library matplotlib sebagai plt.

```
import matplotlib.pyplot as plt
# membuat histogram
plt.hist(df['Price'], bins=10)
# memberi judul dan label pada sumbu x dan y
plt.title('Jumlah Tempat Wisata terhadap Harga Tiket Masuk')
plt.xlabel('Harga Tiket')
plt.ylabel('Jumlah Tempat Wisata')
# menampilkan histogram
plt.show()
```

2. Untuk membuat scatterplot, gunakan plt.scatter(x,y) dengan x dan y adalah dataframe kolom. Kita akan buat scatterplot dengan rating sebagai x, dan harga sebagai y. Kita tidak perlu mengimport ulang library pada ipynb jika sebelumnya kita sudah mengimport.

```
# Membuat scatter plot
plt.scatter(df['Rating'], df['Price'])
# Memberi label pada sumbu x dan y
plt.xlabel('Rating')
plt.ylabel('Harga Tiket')
# Menampilkan scatter plot
plt.show()
```

3. Untuk menggambarkan komposisi dalam persentase kategori tempat wisata di Indonesia, maka kita gunakan piechart, karena piechart cocok untuk membandingkan komposisi persentase dan tabel frekuensi.

```
# Kelompokkan tempat wisata berdasarkan kategori
kategori = df.groupby('Category').size()
# Membuat pie chart dan memberi judul
plt.title('Pie Chart Persentase Kategori Tempat Wisata di Indonesia')
plt.pie(kategori.values, labels=kategori.index, autopct='%1.1f%%')
# Menampilkan pie chart
plt.show()
```

Hasil dari df.groupby('kolom').size() akan terbagi menjadi dua, yaitu index dan values. Values adalah nilai dari frekuensi data. Contohnya pada kasus ini,

Dipindai dengan CamScanner

values adalah 47, 115, 106, dan seterusnya. Sedangkan index adalah nilai dari data. Contohnya pada kasus ini, index adalah "Bahari", "Budaya", "Cagar Alam", dan seterusnya.

4. Untuk melakukannya, kita harus mengelompokkan data berdasarkan kota. Lalu buat barchart dengan plt.bar(x,y) :

```
1. # menghitung jumlah tempat wisata pada tiap kota
2. kota = df.groupby('City').size()
3. # membuat bar chart
4. plt.bar(kota.index, kota.values)
5. # memberi label pada sumbu x dan y
6. plt.title('Jumlah Tempat Wisata Tiap Kota')
7. plt.xlabel('Kota')
8. plt.ylabel('Jumlah Tempat Wisata')
9. # menampilkan bar chart
10. plt.show()
```

Dipilihnya barchart karena dapat menunjukkan perbandingan jumlah (frekuensi) pada kategori tertentu, dalam kasus ini adalah kota.

5. Untuk melakukannya, kita perlu mengambil data tempat wisata di Yogyakarta. Barulah kita bisa membuat histogram dengan x adalah rating, dan y adalah frekuensi dengan cara berikut :

```
1. # Ambil data tempat wisata di Yogyakarta
2. wisata_jogja = df[df['City'] == 'Yogyakarta']
3. # Buat histogram distribusi rating tempat wisata di Yogyakarta
4. plt.hist(wisata_jogja['Rating'], bins=10)
5. # Tambahkan judul dan label sumbu pada grafik
6. plt.title('Jumlah Tempat Wisata terhadap Rating di Yogyakarta')
7. plt.xlabel('Rating')
8. plt.ylabel('Frekuensi')
9. # Tampilkan grafik
10. plt.show()
```

Histogram dipilih karena kolom rating memiliki interval nilai tertentu. Histogram adalah grafik terbaik untuk memvisualisasikan distribusi data untuk interval-interval nilai atau periode waktu. Setiap batang pada histogram merepresentasikan frekuensi data untuk tiap intervalnya.

6. Untuk membuat histogram berat badan bebek jantan dengan bins 15, caranya :

```
1. # Filter hanya data bebek jantan
2. data_bebek_jantan = data_bebek[data_bebek['Gender'] == 'Jantan']
3. # Buat histogram
4. plt.hist(data_bebek_jantan['Berat'], bins=15)
5. # Buat judul dan label sumbu x dan y
6. plt.title('Distribusi Berat Badan Bebek Jantan')
7. plt.xlabel('Berat (kg)')
8. plt.ylabel('Frekuensi')
9. # Tampilkan grafik
10. plt.show()
```

7. Untuk membuat pie chart dengan persentase jenis kelamin bebek yang memiliki berat badan di bawah 2kg, lakukan :

Dipindai dengan CamScanner

```
# Filter hanya data bebek dengan berat badan di bawah 2kg
data_bebek_kurang2kg= data_bebek[data_bebek['Berat'] < 2]
# Hitung persentase jenis kelamin
count_gender = data_bebek_kurang2kg.groupby(['Gender']).size()
# Buat pie chart
plt.pie(count_gender.values, labels=count_gender.index,
autopct='%1.1f%%')
# Tulis judul
plt.title('Jenis Kelamin Bebek dengan Berat Badan di Bawah 2kg')
# Tampilkan grafik
plt.show()
```

8. Untuk menggambarkan scatter plot, maka lakukan :

```
# Buat scatter plot
plt.scatter(data_bebek['Tinggi'], data_bebek['Berat'])
# Tulis judul dan label sumbu x dan y
plt.title('Scatter Plot Tinggi Badan dan Berat Badan Bebek')
plt.xlabel('Tinggi (cm)')
plt.ylabel('Berat (kg)')
# Tampilkan grafik
plt.show()
```

9. Grafik yang paling cocok untuk menggambarkan tinggi dan lebar badan 10 bebek terberat adalah barchart. Kita akan plot berat bebek sebagai x, tinggi dan lebar bebek sebagai y. Grafik barchart cocok karena dapat memvisualisasikan sumbu y dengan lebih dari 1 data.

```
# Urutkan data bebek berdasarkan beratnya secara descending
data_bebek_urut = data_bebek.sort_values(['Berat'], ascending=0)
# Ambil 10 data pertama
data_bebek_urut10 = data_bebek_urut[:10]
# Buat plot dengan x adalah berat dan y adalah tinggi dan lebar
data_bebek_urut10.plot(title="Tinggi dan Lebar Bebek Terhadap
Berat",kind="bar",x="Berat",y=["Tinggi","Lebar"])
```

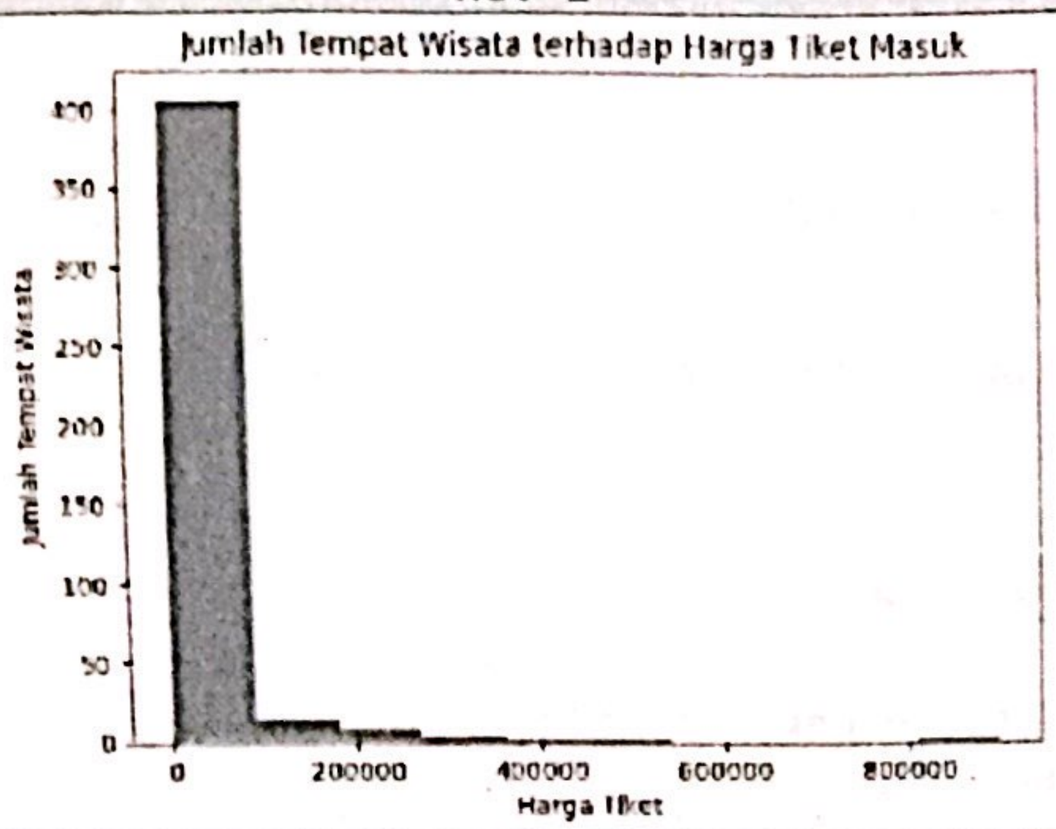
10. Grafik yang paling cocok adalah stacked barchart. Warna pada bebek memiliki variasi unik, yaitu hitam, putih, dan cokelat. Kita dapat menumpuknya dengan stacked barchart.

```
# Kelompokkan data gender dengan warna bebek
kelompok = data_bebek.groupby(['Gender', 'Warna']).size().unstack()
# Buatlah stacked barchart
fig, ax = plt.subplots()
kelompok.plot(kind='bar', stacked=True, ax=ax)
# Tulis judul dan label sumbu x dan y
ax.set_xlabel('Jenis Kelamin')
ax.set_ylabel('Jumlah Bebek')
ax.set_title('Jumlah Bebek Berdasarkan Jenis Kelamin dan Warna')
# Tampilkan grafik
plt.show()
```

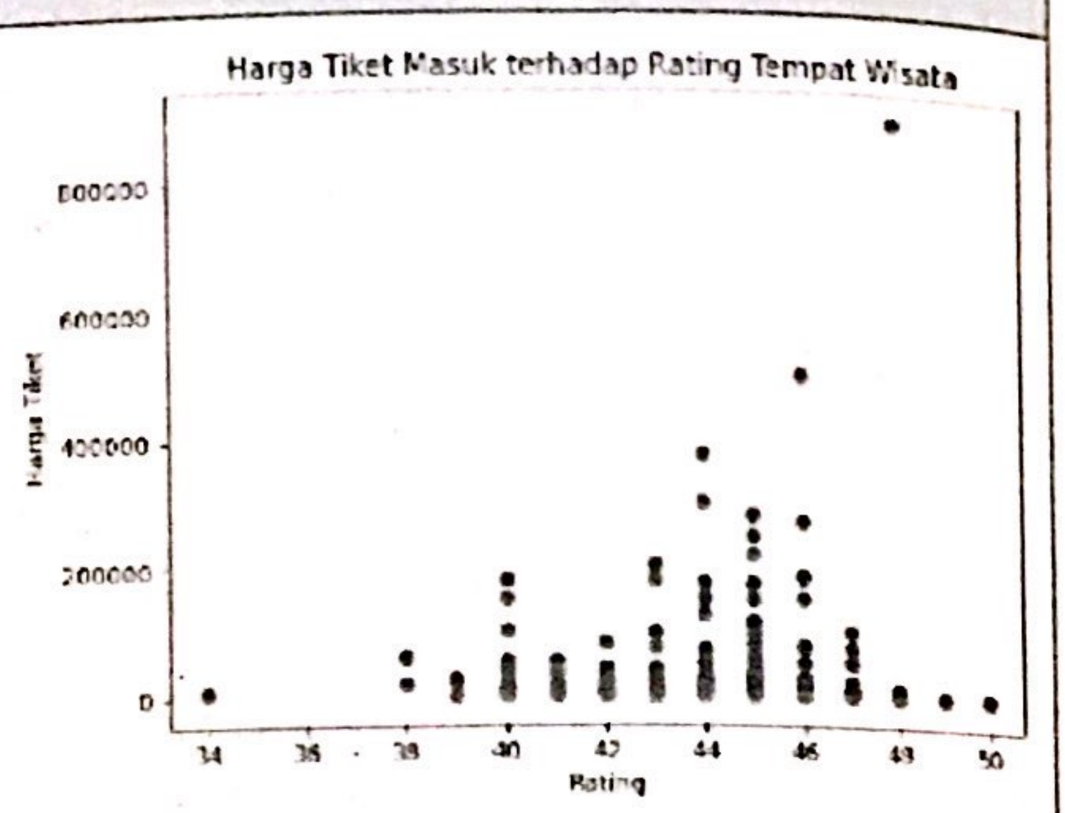

Dipindai dengan CamScanner

HASIL VISUALISASI

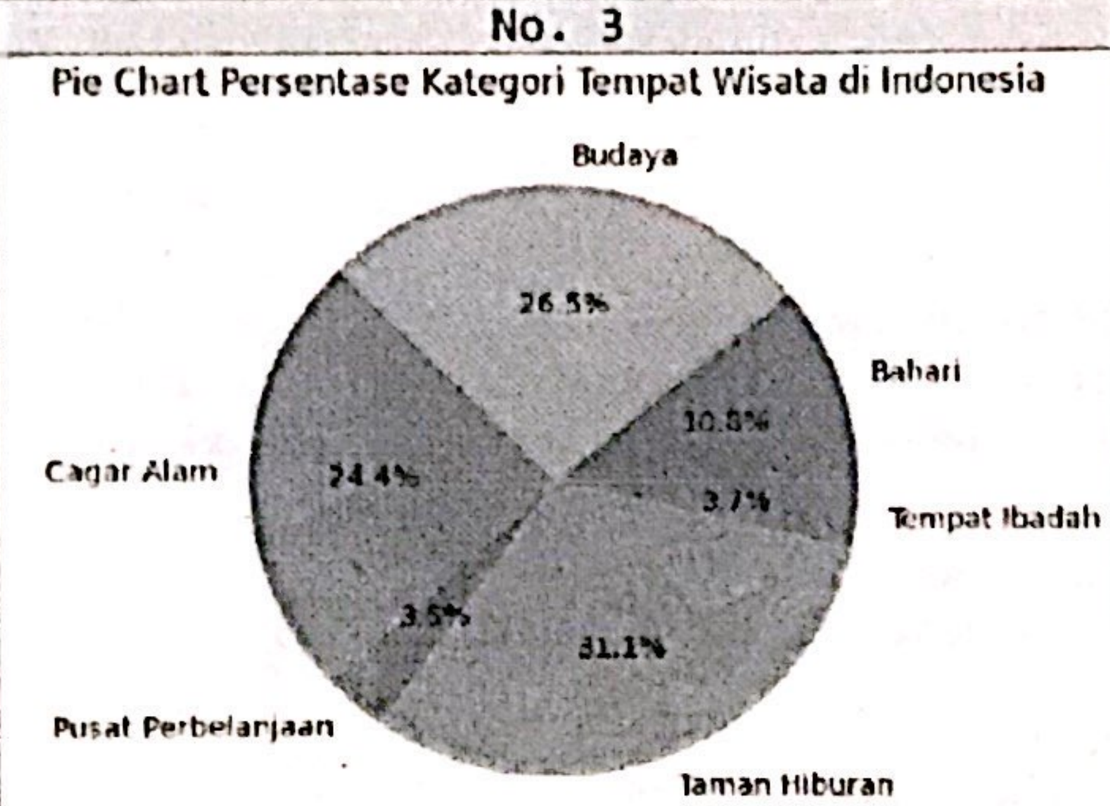
No. 1



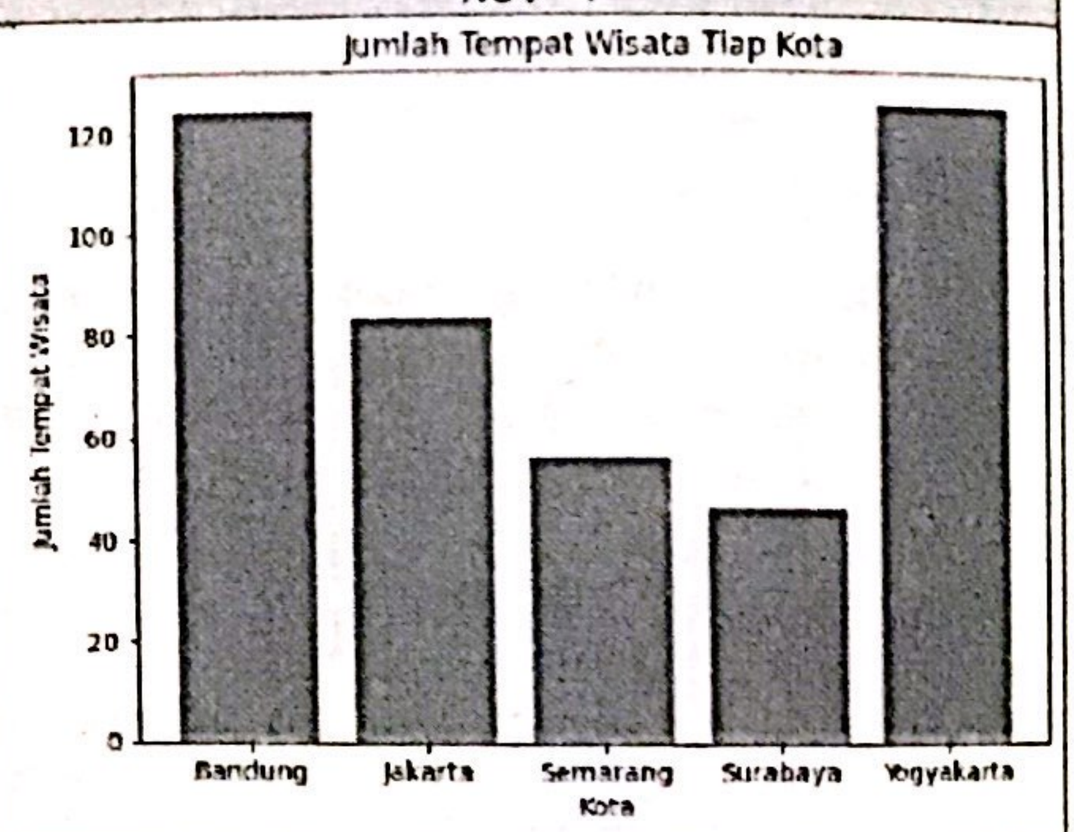
No. 2



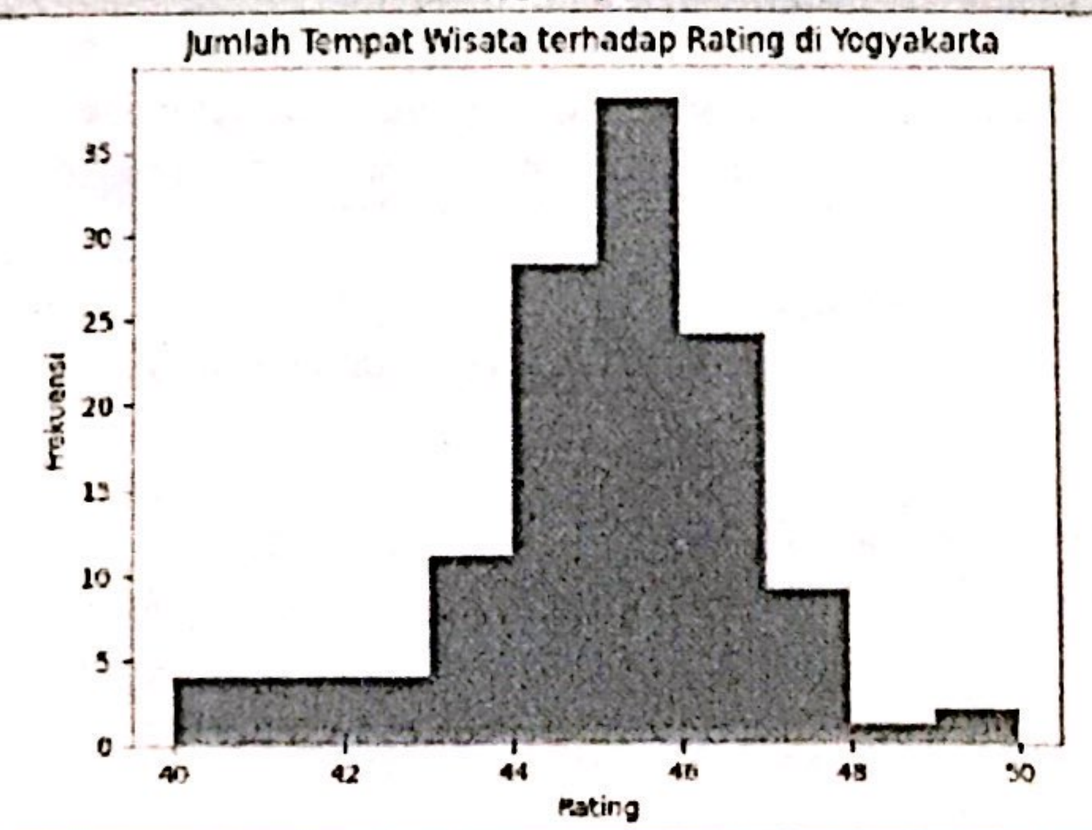
No. 3



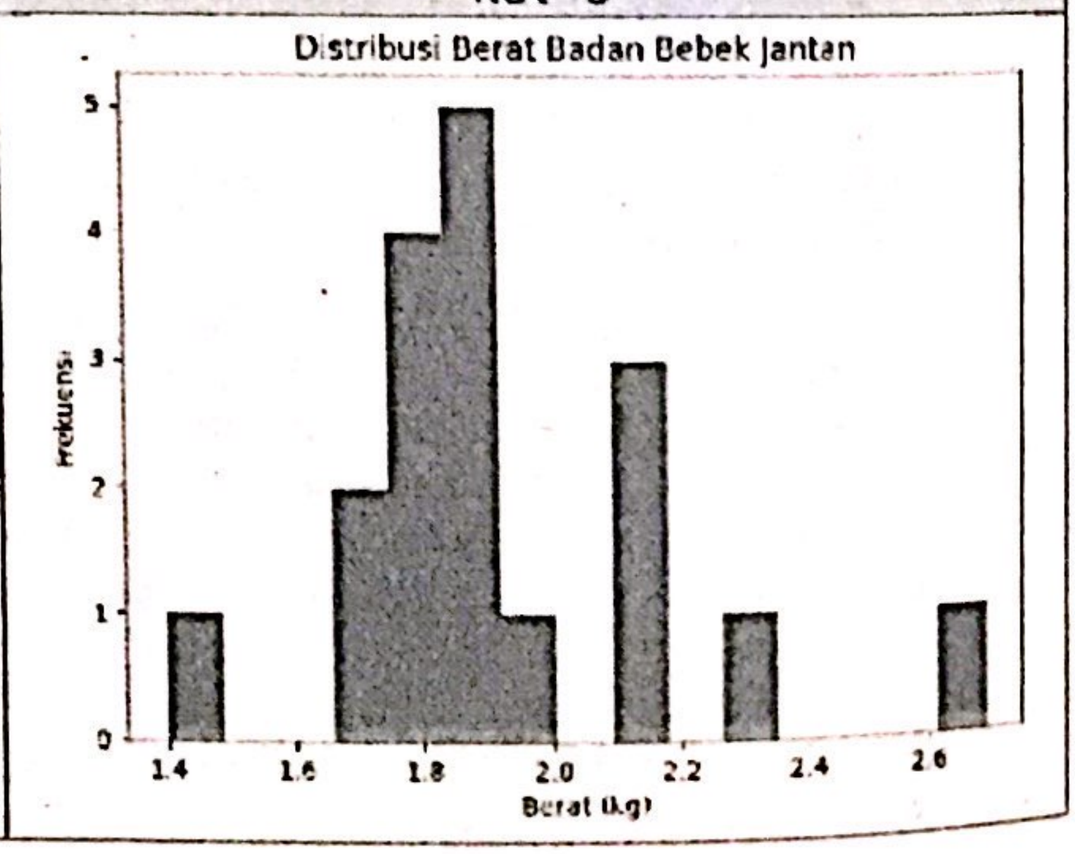
No. 4



No. 5

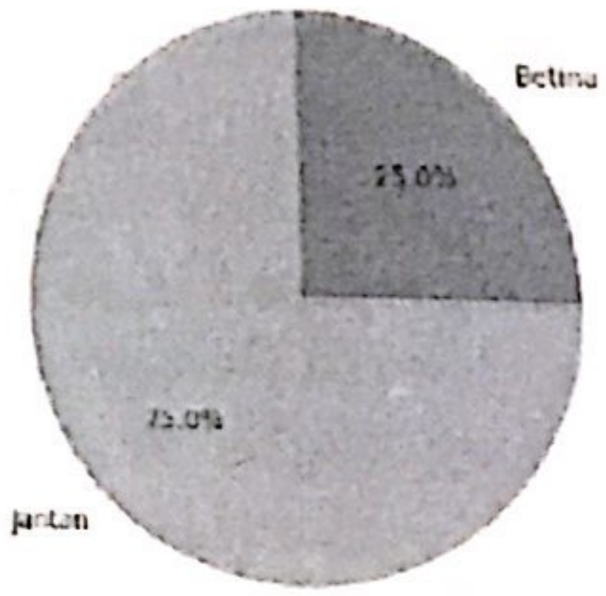


No. 6



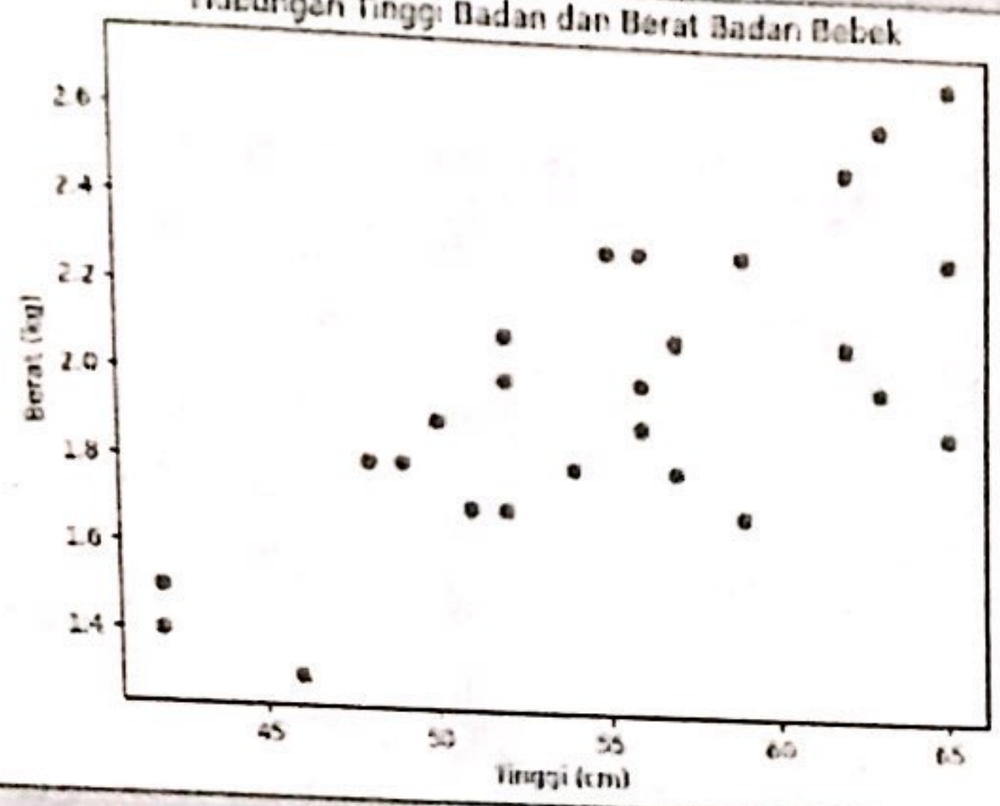
No. 7

Persentase Jenis Kelamin Bebek dengan Berat Badan di Bawah 2kg



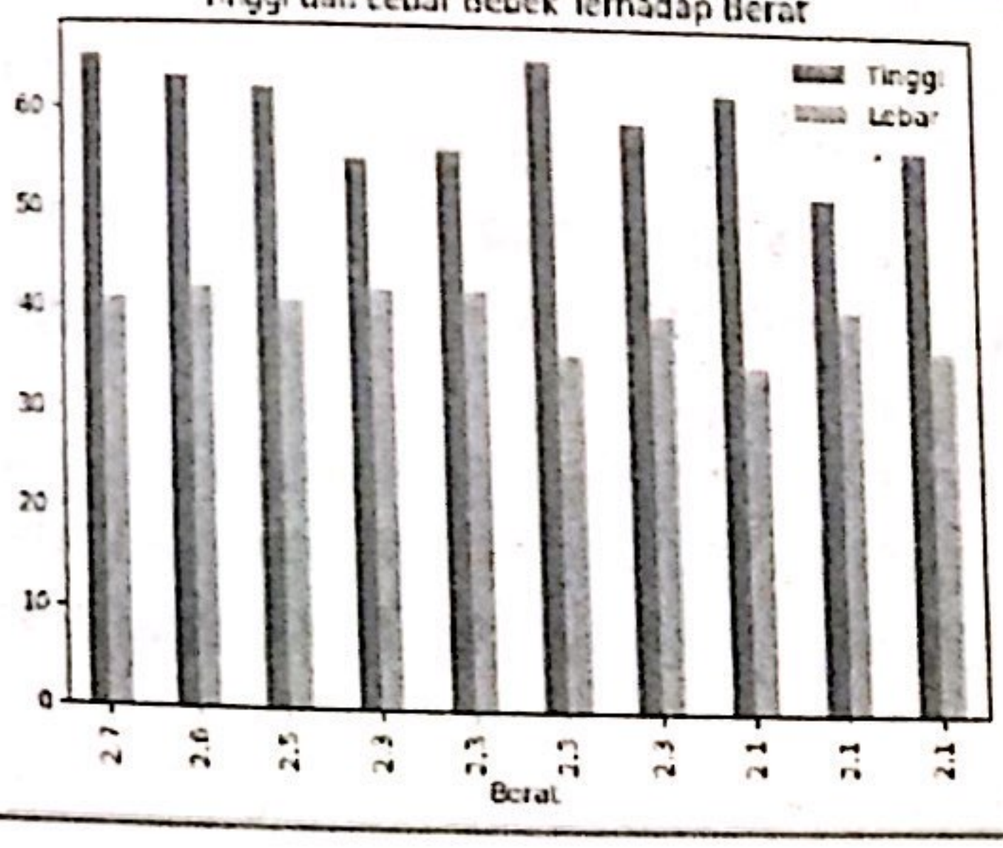
No. 8

Hubungan Tinggi Badan dan Berat Badan Bebek



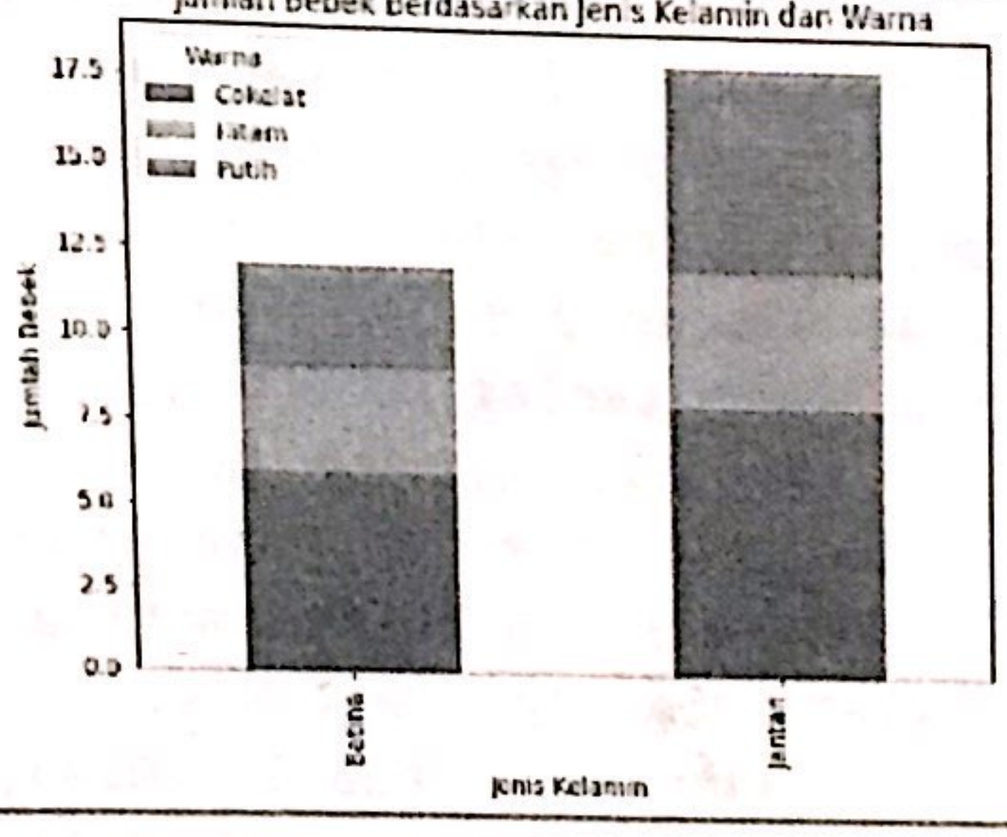
No. 9

Tinggi dan Lebar Bebek Terhadap Berat



No. 10

Jumlah Bebek Berdasarkan Jenis Kelamin dan Warna



Versi berwarna :

Dipindai dengan CamScanner