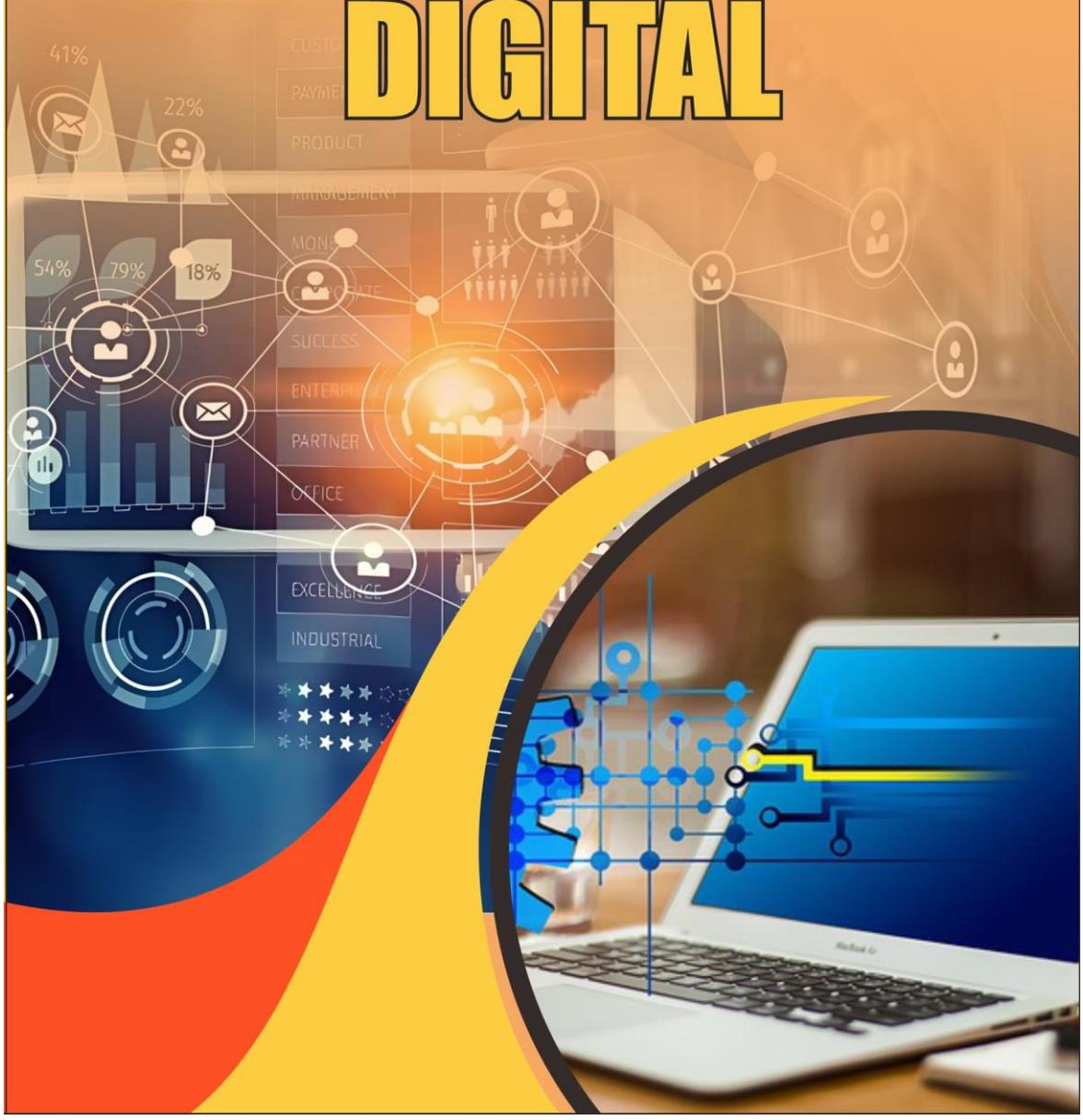




TRANSFORMASI DIGITAL



TRANSFORMASI DIGITAL

Nalar Istiqomah, M.Kom



Tahta Media Group

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

TRANSFORMASI DIGITAL

Penulis:

Nalar Istiqomah, M.Kom

Desain Cover:

Tahta Media

Editor:

Tahta Media

Proofreader:

Tahta Media

Ukuran:

viii, 315, Uk: 15,5 x 23 cm

ISBN: 978-623-147-262-5

Cetakan Pertama:

Desember 2023

Hak Cipta 2023, Pada Penulis

Isi diluar tanggung jawab percetakan

Copyright © 2023 by Tahta Media Group

All Right Reserved

Hak cipta dilindungi undang-undang
Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa izin tertulis dari Penerbit.

PENERBIT TAHTA MEDIA GROUP
(Grup Penerbitan CV TAHTA MEDIA GROUP)
Anggota IKAPI (216/JTE/2021)

PRAKATA

Puji dan syukur penulis panjatkan kepada Allah SWT karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan Buku Transformasi Digital. Secara umum, transformasi digital adalah proses pemanfaatan teknologi digital untuk membangun cara-cara baru dalam berbagai aktivitas manusia dan/atau menyelesaikan persoalan yang dihadapi. Secara lebih spesifik, buku ini akan membahas:

1. Pengenalan bentuk-bentuk transformasi dan disrupsi yang terjadi pada masa kini.
2. Mengidentifikasi masalah-masalah khas era digital serta memberikan gagasan solusi untuk masalah tersebut dengan system thinking dan critical thinking sebagai tools.
3. Pengenalan proses pengembangan program komputer secara sederhana untuk melatih computational thinking skills. Buku ini menggunakan bahasa pemrograman Python untuk membuat program komputer.

Buku ini disertai dengan contoh pemrograman sederhana beserta latihan soal untuk menguji pemahaman terhadap materi yang diberikan. Di bagian akhir buku, diberikan kunci jawaban untuk memeriksa jawaban pembaca.

Meskipun penulis telah berusaha sebaik mungkin untuk menyusun buku ini, penulis menyadari bahwa buku ini masih banyak kekurangan. Maka dari itu, penulis sangat mengharapkan kritik dan saran untuk menyempurnakan segala kekurangan buku ini. Semoga buku ini bisa memberikan manfaat bagi pembaca dan pihak lainnya.

Penulis

DAFTAR ISI

| | |
|--|-----------|
| Prakata | iv |
| Daftar Isi..... | v |
| Pendahuluan | 1 |
| Bab 1 Konsep Transformasi Digital..... | 2 |
| A. Materi Pembelajaran | 2 |
| 1. Revolusi Industri | 3 |
| 2. Unsur Utama Perkembangan Revolusi Industri 4.0 | 7 |
| 3. Konsep <i>Disruptive Innovation</i> | 9 |
| 4. <i>Skill</i> yang diperlukan di Revolusi Industri 4.0 | 11 |
| B. Latihan Soal | 13 |
| C. Rangkuman | 13 |
| Bab 2 <i>Computational Thinking</i> | 15 |
| A. Materi Pembelajaran | 15 |
| 1. Pengertian <i>Computational Thinking</i> | 15 |
| 2. Pengantar Algoritme..... | 19 |
| 3. Pengenalan Bahasa Python..... | 28 |
| B. Latihan Soal | 35 |
| C. Rangkuman | 36 |
| Bab 3 Tipe Data, Variabel, Operasi Input Output, Dan Operator Dasar | 37 |
| A. Materi Pembelajaran | 37 |
| 1. Tipe Data dan Variabel..... | 37 |
| 2. Cara Membuat Variabel | 38 |
| 3. Tipe Data Pada Python | 41 |
| 4. Operasi Input Output | 44 |
| 5. Operator dasar | 49 |
| B. Latihan Soal | 54 |
| C. Rangkuman | 56 |
| Bab 4 Boolean Value, Operator Logika Dan Percabangan | 57 |
| A. Materi Pembelajaran | 57 |
| 1. Boolean Value | 57 |
| 2. Operator Logika..... | 61 |

| | |
|---|------------|
| 3. Kondisional atau Percabangan | 64 |
| B. Latihan Soal | 73 |
| C. Rangkuman | 77 |
| Bab 5 Perulangan, Tipe Data List, Dan Operasi Dalam List | 78 |
| A. Materi Pembelajaran | 78 |
| 1. Perulangan | 78 |
| 2. Perulangan Bersarang/Bertingkat | 93 |
| 3. Tipe Data List | 96 |
| 4. Operasi Dalam List | 103 |
| B. Latihan Soal | 104 |
| C. Rangkuman | 105 |
| Bab 6 Penggunaan Pemrograman Modular (Fungsi) | 107 |
| A. Materi Pembelajaran | 107 |
| 1. Fungsi | 107 |
| 2. Fungsi Yang Memanggil Dirinya Sendiri (Rekursif) | 115 |
| B. Latihan Soal | 122 |
| C. Rangkuman | 123 |
| Bab 7 Penggunaan Tuple, Dictionary, Dan Pemrosesan Data | 125 |
| A. Materi Pembelajaran | 125 |
| 1. Tipe Data Tuple | 125 |
| 2. Tipe Data Dictionary | 132 |
| B. Latihan Soal | 140 |
| C. Rangkuman | 141 |
| Bab 8 Budaya Digital..... | 143 |
| A. Materi Pembelajaran | 143 |
| 1. Komunikasi Sosial..... | 147 |
| 2. Kolaborasi | 150 |
| 3. Keamanan Informasi | 152 |
| 4. Etika..... | 156 |
| 5. Privasi | 160 |
| B. Latihan Soal | 162 |
| C. Rangkuman | 162 |
| Bab 9 System Thinking | 164 |
| A. Materi Pembelajaran | 164 |
| 1. Sistem | 165 |

| | | |
|----|--|------------|
| 2. | <i>System Thinking</i> | 176 |
| 3. | Esensi Berpikir Sistem | 179 |
| 4. | Dimensi System Thinking | 180 |
| 5. | Prinsip – Prinsip <i>System Thinking</i> | 186 |
| 6. | Metodologi Pemodelan <i>System Thinking</i> | 187 |
| B. | Rangkuman | 190 |
| | Bab 10 Critical Thinking | 192 |
| A. | Materi Pembelajaran | 192 |
| 1. | Manfaat <i>Critical Thinking</i> | 194 |
| 2. | Cara Membentuk <i>Critical Thinking</i> | 195 |
| 3. | Karakteristik <i>Critical Thinking</i> | 197 |
| 4. | Karakteristik Pemikir Kritis | 200 |
| 5. | Faktor Penghambat <i>Critical Thinking</i> | 201 |
| B. | Latihan Soal | 203 |
| C. | Rangkuman | 203 |
| | Bab 11 IoT: Arsitektur, Protokol, Dan Teknologi, Serta Penerapannya | 205 |
| A. | Materi Pembelajaran | 205 |
| 1. | Sejarah <i>Internet of Things</i> | 205 |
| 2. | Cara Kerja IoT | 207 |
| 3. | Unsur – Unsur <i>Internet of Things</i> | 207 |
| 4. | Arsitektur IoT | 209 |
| 5. | Contoh Arsitektur IoT – <i>Intelligent Lighting</i> | 214 |
| 6. | Manfaat IoT | 217 |
| B. | Latihan Soal | 218 |
| C. | Rangkuman | 218 |
| | Bab 12 Kecerdasan Buatan Dan Penerapannya | 221 |
| A. | Materi Pembelajaran | 221 |
| 1. | Komponen Kecerdasan Buatan | 223 |
| 2. | Penggunaan AI | 224 |
| 3. | Algoritme AI: Fuzzy Logic | 226 |
| 4. | Dasar Logika Fuzzy..... | 230 |
| 5. | Proses Sistem Kontrol Logika Fuzzy | 232 |
| 6. | Contoh Perhitungan Fuzzy | 234 |
| 7. | Penerapan Fuzzy..... | 238 |

| | |
|---|------------|
| B. Latihan Soal | 239 |
| C. Rangkuman | 240 |
| Bab 13 MACHINE LEARNING DAN PENERAPANNYA | 243 |
| A. Materi Pembelajaran | 243 |
| 1. Klasifikasi <i>Machine Learning</i> | 246 |
| 2. Aplikasi <i>Machine Learning</i> | 248 |
| 3. <i>Machine Learning Lifecycle</i> | 251 |
| 4. Algoritme Dasar | 258 |
| B. Latihan Soal | 268 |
| C. Rangkuman | 269 |
| Bab 14 Big Data | 271 |
| A. Materi Pembelajaran | 271 |
| 1. Perkembangan <i>Big Data</i> (BD)..... | 272 |
| 2. Tahapan Pengelolaan Big Data | 275 |
| 3. Ekosistem Big Data | 278 |
| 4. Arsitektur Big Data..... | 281 |
| 5. Key Roles Kunci Sukses Proyek Analitik | 283 |
| 6. Introduction to Hadoop..... | 285 |
| B. Rangkuman | 289 |
| Kunci Jawaban | 291 |
| Daftar Pustaka | 313 |
| Biodata Penulis | 314 |
| Sinopsis | 315 |

PENDAHULUAN

DESKRIPSI SINGKAT

Transformasi Digital merupakan mata kuliah yang melatih *system thinking*, *critical thinking* dan *computational thinking* untuk membangun cara-cara baru dalam berbagai aktivitas manusia dan/atau menyelesaikan persoalan yang dihadapi. Mata kuliah ini membahas pengenalan bentuk-bentuk transformasi dan disrupsi yang terjadi pada masa kini. Mata kuliah ini juga membahas masalah-masalah khas era digital serta melatih mahasiswa agar bisa memberikan gagasan solusi untuk masalah tersebut dengan *system thinking* dan *critical thinking* sebagai *tools*. Dikenalkan pula proses pengembangan program komputer secara sederhana untuk melatih *computational thinking skills* menggunakan bahasa pemrograman Python. Selain itu, dikenalkan pula teknologi yang ada pada saat ini seperti IoT, Artificial Intelligence (AI) atau Kecerdasan Buatan, Machine Learning serta Big Data.

RELEVANSI

Mata kuliah ini merupakan pengetahuan dasar yang dibutuhkan mahasiswa untuk menghadapi era revolusi industri 4.0. Dimana pada masa ini, ukuran data tidak lagi ratusan, atau ribuan, tapi bisa jadi ratusan ribu bahkan tak terhingga (Big Data). Oleh karena itu sangat dibutuhkan kemampuan pemrograman untuk mengolah data mulai dari praproses data, kemudian mengolahnya untuk *forecasting*, klasifikasi, *clustering*, dan lain sebagainya. Selain itu, pada masa ini teknologi yang ada telah banyak berkembang, sehingga mahasiswa perlu mengetahui dan mengidentifikasi teknologi mana yang bisa digunakan untuk masalah tertentu.

TUJUAN INSTRUKSIONAL

Diharapkan mahasiswa dapat merelasikan teknologi TIK sebagai *tools* dalam *project* yang dikerjakannya, meskipun tidak sampai pada tahapan implementasi (*coding*).

BAB 1

KONSEP TRANSFORMASI

DIGITAL

A. MATERI PEMBELAJARAN

Transformasi digital merupakan tahap ketiga dari pemanfaatan teknologi digital. Adapun tahapan dalam pemanfaatan teknologi digital adalah:

1. Kompetensi digital (*e-skill*).

Pada tahap ini diperlukan pengetahuan tentang *how to know* dan *how to understand* tentang teknologi digital. Ada 6 kompetensi digital yang harus dimiliki yaitu kemampuan beradaptasi, eksperimental, inovasi, *agility*, kreativitas dan *networking*.

2. Penggunaan digital.

Pada tahapan ini kita bukan hanya tahu dan mengerti. Tapi juga bisa menggunakan dan mengoptimalkan teknologi digital. Penggunaan digital erat kaitannya dengan kecakapan kita dalam menggunakan teknologi informasi dan komunikasi berdasarkan kaidah atau aturan yang berlaku dengan menyadari implikasinya pada masalah yang kita hadapi.

3. Transformasi digital.

Tahap ini berarti kita bukan hanya bisa menggunakan, tapi juga bisa membuat inovasi baru menggunakan teknologi digital dalam domain (bidang) tertentu.

Transformasi digital adalah bagian proses teknologi yang berkaitan dengan perubahan penerapan teknologi digital dalam semua aspek kehidupan masyarakat. Transformasi digital muncul karena adanya revolusi industri. Maka modul ini akan dimulai dengan pembahasan revolusi industri terlebih dahulu.

1. Revolusi Industri

Revolusi industri adalah perubahan besar terhadap cara manusia dalam mengolah sumber daya dan memproduksi barang. Pada revolusi industri, terjadi perubahan besar-besaran di berbagai bidang seperti pertanian, manufaktur, pertambangan, transportasi, teknologi dan lain sebagainya. Perubahan tersebut ikut berdampak pada kondisi sosial, ekonomi, dan budaya di dunia. Berikut adalah gambaran dari revolusi industri.

| | |
|---|---|
|  <p>REVOLUSI INDUSTRI 1.0 Abad ke-18</p> <p>Mesin uap menggantikan tenaga manusia di bidang industri tekstil. Mesin uap juga digunakan pada kapal laut.</p> |  <p>REVOLUSI INDUSTRI 2.0 Abad ke-19</p> <p>Tenaga listrik digunakan dalam industri dalam rangka produksi massal.</p> |
|  <p>REVOLUSI INDUSTRI 3.0 Abad ke-20</p> <p>Penggunaan mesin otomatis serta teknologi informasi.</p> |  <p>REVOLUSI INDUSTRI 4.0 Masa Kini</p> <p>Mesin terintegrasi dengan jaringan internet</p> |

Gambar 1 Revolusi Industri

Revolusi industri 1.0

Revolusi industri yang pertama terjadi pada abad ke-18. Ditandai dengan penemuan mesin uap oleh James Watt pada tahun 1776. Mesin uap ini digunakan dalam proses produksi barang dan transportasi. Saat itu, di Inggris, peralatan untuk menenun kain bergantung pada tenaga manusia dan hewan. Setelah ditemukan mesin uap, mesin tersebut kemudian digunakan sebagai alat tenun mekanis pertama yang dapat meningkatkan produktivitas industri tekstil.

Selain itu, mesin uap digunakan pada bidang transportasi. Transportasi internasional pada masa itu adalah transportasi laut yang masih menggunakan kapal bertenaga angin. Namun, angin tidak dapat sepenuhnya dapat diandalkan karena angin tidak bisa diprediksi. Bisa jadi

angin bertiup dari arah yang berlawanan atau bahkan tidak ada angin sama sekali. Sejak mesin uap ditemukan, kapal bertenaga angin digantikan dengan kapal bertenaga uap yang jauh lebih efisien dan lebih murah. Dengan mesin uap tersebut, kapal dapat berlayar selama 24 jam penuh jika mesin uap tetap didukung dengan kayu dan batu bara yang cukup.

Revolusi industri ini memungkinkan bangsa Eropa mengirim kapal perang ke seluruh penjuru dunia dalam waktu yang jauh lebih singkat. Kemudian, negara-negara imperialis di Eropa mulai menjajah kerajaan-kerajaan di Afrika dan Asia. Selain penjajahan, terdapat dampak lain dari revolusi industri pertama, yaitu pencemaran lingkungan akibat asap mesin uap dan limbah-limbah pabrik lainnya.

Revolusi industri 2.0

Revolusi industri 2.0 terjadi di awal abad ke-19. Revolusi industri ini ditandai dengan penemuan tenaga listrik yang menggantikan mesin uap.

Di akhir 1800-an, mobil mulai diproduksi secara massal. Produksi massal ini memakan waktu yang lama karena setiap mobil harus dirakit dari awal hingga selesai di titik yang sama oleh seorang perakitan mobil. Artinya, untuk merakit banyak mobil, proses perakitan harus dilakukan oleh banyak orang yang merakit mobil dalam waktu yang bersamaan.

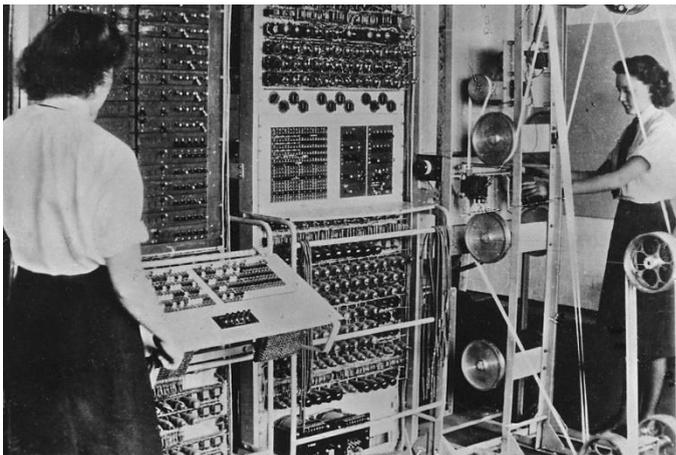
Revolusi terjadi dengan terciptanya "lini produksi" atau *assembly line* yang menggunakan "ban berjalan" atau *conveyor belt* pada tahun 1913. Penemuan ini mengakibatkan proses produksi berubah total karena untuk menyelesaikan satu mobil, tidak diperlukan satu orang untuk merakit dari awal hingga akhir. Para perakitan mobil dilatih untuk menjadi spesialis yang mengurus satu bagian saja. Mobil dirakit pada tahap pertama oleh seorang perakitan mobil, kemudian dioper ke bagian selanjutnya menggunakan "ban berjalan", begitu seterusnya hingga tahap terakhir. Selain itu, para perakitan mobil melakukan pekerjaannya dengan bantuan alat-alat yang menggunakan tenaga listrik yang jauh lebih mudah dan murah daripada tenaga uap.

Revolusi industri kedua ini juga berdampak pada kondisi militer pada perang dunia II. Ribuan tank, pesawat, dan senjata diciptakan dari pabrik-pabrik yang menggunakan lini produksi dan ban berjalan. Hal ini terjadi karena adanya produksi massal (*mass production*).

Revolusi industri 3.0

Pada revolusi industri kedua, manusia masih berperan sangat penting dalam proses produksi berbagai macam jenis barang. Tetapi, pada revolusi industri yang ketiga, manusia tidak lagi memegang peranan penting. Setelah revolusi ini, abad industri pelan-pelan berakhir dan abad informasi dimulai.

Jika revolusi pertama dipicu oleh mesin uap, revolusi kedua dipicu oleh ban berjalan dan listrik, revolusi ketiga ini dipicu oleh mesin yang dapat bergerak dan berpikir secara otomatis, yaitu komputer dan robot. Salah satu komputer pertama yang dikembangkan di era perang dunia II adalah komputer Colossus, mesin yang digunakan untuk memecahkan kode buatan Nazi Jerman. Bentuk komputer Colossus berbeda dengan komputer yang kita temui saat ini. Komputer Colossus berukuran sangat besar, tidak memiliki RAM dan tidak bisa menerima perintah dari manusia melalui keyboard. Komputer purba tersebut hanya menerima perintah melalui pita kertas yang membutuhkan daya listrik sangat besar, yaitu 8.500 watt.



*Gambar 2 Komputer Colossus (sumber:
<https://www.britannica.com/technology/Colossus-computer>)*

Namun, kemajuan teknologi berkembang luar biasa pesat setelah perang dunia kedua selesai. Dalam pengembangan tersebut, kemudian

BAB 2

COMPUTATIONAL THINKING

A. MATERI PEMBELAJARAN

1. Pengertian Computational Thinking

Computational thinking adalah cara berpikir untuk menyelesaikan masalah (*problem solving*) dengan cara menguraikannya menjadi beberapa tahapan yang efektif, efisien, dan menyeluruh. Tahapan *computational thinking* ini meliputi: *decomposition*, *pattern recognition*, *abstraction*, dan *algorithms* yang merupakan beberapa konsep dasar ilmu komputer.

Jeanette Wing memperkenalkan istilah *computational thinking* pada Maret 2006. Menurutnya, *computational thinking* melibatkan pemecahan masalah, perancangan sistem, dan memahami perilaku manusia dengan berkaca pada konsep-konsep dasar ilmu komputer.

Apa saja tahapan *computational thinking*? seperti yang telah dijelaskan sebelumnya, *computational thinking* melibatkan 4 tahap utama, yakni:

- a. *Decomposition*, merupakan pembagian masalah menjadi bagian-bagian yang lebih kecil atau sederhana.
"Jika sebuah masalah tidak terurai, maka jauh lebih sulit untuk dipecahkan. Berurusan dengan banyak tahapan yang berbeda sekaligus jauh lebih sulit daripada memecahkan masalah menjadi sejumlah masalah yang lebih kecil dan menyelesaikan setiap masalah, satu per satu." – BBC Bitesize
- b. *Pattern recognition*, yakni mencari atau mengenali kesamaan pola dalam maupun antar masalah yang ingin dipecahkan.
- c. *Abstraction*, melihat permasalahan secara mendasar sehingga dapat melihat jangkauan luas yang lebih penting dan mengabaikan detail kecil yang sebetulnya kurang relevan.

"Jika kita tidak mengabstrakan diri kita mungkin berakhir dengan solusi yang salah untuk masalah yang kita coba pecahkan." – BBC Bitesize

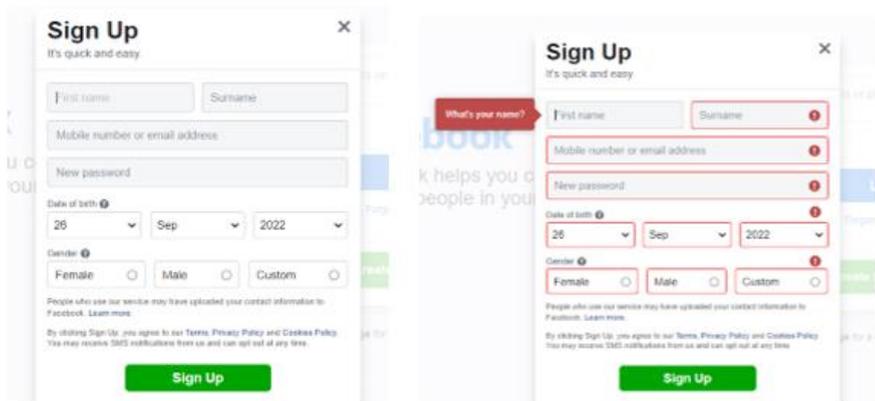
- d. *Algorithm*, mengembangkan sistem, sekuen, atau langkah-langkah solusi yang dapat diterapkan secara menyeluruh terhadap pola yang sama sehingga lebih efektif dan efisien. Dalam ilmu komputer, tahap ini disebut dengan *coding*.

Dapat dikatakan pula bahwa apa itu *computational thinking* atau pemikiran komputasi merupakan metode untuk menyelesaikan masalah melalui tahapan-tahapan berpikir dalam merancang pengembangan sistem atau aplikasi komputer. Ya, karena pada awalnya, pemikiran ini adalah pemikiran yang digunakan untuk memecahkan permasalahan ilmu komputer. Dalam membuat kode program, *programmer* harus bisa menerjemahkan cara berpikir manusia yang abstrak, menjadi terstruktur agar dapat dimengerti oleh komputer.

Meskipun begitu, pemikiran komputasi dapat diimplementasikan terhadap permasalahan lain di luar ilmu komputer pula. Hal ini dapat terlihat dari definisi berbeda dari salah satu pemikir utama *computational thinking*, yakni Jeannette M. Wing. Pada tahun 2011, Wing mengubah definisi pemikiran komputasi menjadi: proses berpikir yang merumuskan masalah dan solusinya, sehingga solusi tersebut direpresentasikan dalam bentuk yang dapat dilakukan secara efektif oleh agen pengolah informasi.

Contoh *computational thinking* dalam pemrograman aplikasi

Kebutuhan aplikasi akan banyak menimbulkan masalah yang harus dipecahkan agar aplikasi tersebut dapat bekerja dan digunakan. Salah satu contohnya adalah bagaimana input dari *user* harus diverifikasi validitasnya. Misalnya, suatu form pendaftaran yang memiliki isian wajib diisi, jika tidak diisi aplikasi tidak dapat memprosesnya.



Gambar 6 Contoh form pendaftaran

Langkah computational thinkingnya yaitu

- 1) *Decomposition*, bagaimana cara mengecek suatu form telah diisi oleh *user* atau belum? Manusia bisa saja langsung tahu formnya sudah diisi atau belum dengan sekali lihat. Tapi, komputer tidak bisa seperti itu. Komputer harus mengecek satu-persatu kolom isian. Misal, kita mulai dari kolom isian “Nama” terlebih dahulu. Kita memberikan perintah ini pada sistem:
 Jika kolom nama tidak diisi, Maka: “beri peringatan dan hentikan proses pendaftaran aplikasi”
 Jika kolom nama diisi, Maka: “lanjutkan proses pendaftaran aplikasi”.
 Sehingga, jika kolom nama tidak diisi maka aplikasi akan menghentikan proses pendaftaran, sebaliknya jika kolom nama diisi, maka proses pendaftaran aplikasi akan dilanjutkan.
- 2) *Pattern recognition*, di dalam form pendaftaran banyak kolom isian lain yang wajib untuk diisi juga seperti nomor hp, alamat email, password dan lain sebagainya. Dengan *pattern recognition*, kita bisa mengenali bahwa untuk memeriksa kolom lain sudah diisi atau belum, dapat dilakukan dengan menggunakan algoritme yang sama dengan algoritme pada langkah 1.
- 3) *Abstraction*, ketika kita membuat kode form isian pendaftaran, kita tidak perlu memperhatikan bagaimana cara penyajiannya, UI dan UXnya dan

BAB 3

TIPE DATA, VARIABEL, OPERASI INPUT OUTPUT, DAN OPERATOR DASAR

A. MATERI PEMBELAJARAN

Pada bab sebelumnya, kita telah melakukan instalasi Python dan membuat program sederhana. Pada bab ini, kita akan membahas lebih lanjut tentang pemrograman dengan Python, yaitu tipe data, variabel, operasi input output serta operator dasar pada Python.

1. Tipe Data dan Variabel

Apa itu variabel? Anggap saja variabel adalah sebuah keranjang, tempat kita bisa memasukkan sesuatu di dalamnya, yaitu data. Di Python, kita bisa memasukkan tipe data apa saja ke dalam keranjang (yakni variabel). Tipe data adalah jenis dari suatu data. Setiap data memiliki nilai, dan setiap nilai memiliki jenis. Ada data-data yang bertipe angka, ada pula yang bertipe huruf/karakter, ada juga yang bertipe benar/salah dan sebagainya.

Sebagai ibarat, jika variabel adalah keranjang, maka data adalah buahnya sedangkan tipe data adalah jenis buah yang akan kita masukkan ke dalam keranjang tersebut. Gambar di bawah ini adalah ilustrasi dasar bagaimana hubungan sebuah variabel dan tipe data.



Gambar 6 Gambaran variabel, data dan tipe data (sumber: <https://padiumkm.id/product/buah-buahan/6482ed0d174b2142ed181bd4>)

Kita bisa lihat bahwa di dalam gambar di atas, terdapat banyak keranjang dan banyak buah. Setiap keranjang tertentu digunakan untuk menyimpan jenis buah tertentu. Sehingga bisa kita tarik kesimpulan bahwa:

- a. Keranjang merepresentasikan variabel.
- b. Buah merepresentasikan data.
- c. Dan jenis-jenis buah tersebut merepresentasikan tipe data.

2. Cara Membuat Variabel

Membuat variabel pada Python itu sangat mudah. Kita hanya perlu menuliskan nama variabel lalu diikuti oleh nilai yang kita inginkan. Misal,

```
nama = 'Nalar Istiqomah'  
usia = 28  
pekerjaan = 'IRT'  
print('Nama: ', nama)  
print('Usia: ', usia)  
print('Pekerjaan: ', pekerjaan)
```

Jika dieksekusi, program tersebut akan menghasilkan output sebagai berikut

```
D:\New folder (2)>latihan.py
Nama: Nalar Istiqomah
Usia: 28
Pekerjaan: IRT
```

Pada program di atas kita membuat 3 variabel yaitu

- 1) Nama
- 2) Usia
- 3) Pekerjaan

Masing-masing variabel diberikan sebuah nilai sebagai berikut

- 1) Variabel nama memiliki nilai 'Nalar Istiqomah'
- 2) Variabel usia memiliki nilai 28
- 3) Variabel pekerjaan memiliki nilai 'IRT'

Setelah diberi nilai, lalu ditampilkan masing-masing variabelnya menggunakan perintah `print()`.

Aturan Penamaan Variabel

Secara umum, kita bisa membuat nama variabel apa saja yang kita mau dalam Python. Akan tetapi, ada beberapa aturan dan pengecualian. Berikut adalah aturan-aturannya

- 1) Nama variabel hanya boleh diawali huruf atau *underscore*
- 2) Nama variabel tidak boleh diawali oleh angka

```
>>> 6nama= 'Nalar Istiqomah'
      File "<stdin>", line 1
        6nama= 'Nalar Istiqomah'
          ^
SyntaxError: invalid syntax
```

- 3) Nama variabel hanya bisa terdiri dari karakter *alpha-numeric* dan *underscore* (A-z, 0-9 dan `_`)

```
>>> nam*a= 'Nalar Istiqomah'
      File "<stdin>", line 1
        nam*a= 'Nalar Istiqomah'
          ^
SyntaxError: cannot assign to operator
>>> nam>a= 'Nalar Istiqomah'
      File "<stdin>", line 1
        nam>a= 'Nalar Istiqomah'
          ^
SyntaxError: cannot assign to comparison
```

BAB 4

BOOLEAN VALUE, OPERATOR LOGIKA DAN PERCABANGAN

A. MATERI PEMBELAJARAN

Pada bab sebelumnya, kita telah membahas berbagai tipe data, salah satunya adalah tipe data Boolean. Kita juga telah membahas tentang berbagai macam operator di Python. Pada pertemuan ini, kita akan membahas tipe data boolean dan operator logika lebih detail. Kita juga akan membahas kondisional atau percabangan dalam pemrograman Python

1. Boolean Value

Dari seluruh tipe data yang ada di dalam bahasa pemrograman Python, Boolean merupakan tipe data yang paling sederhana. Meskipun begitu, **Boolean** sangat penting dan selalu ada di setiap bahasa pemrograman komputer. Variabel ini pertama kali diperkenalkan oleh George Boole.

Pada Python, tipe data Boolean memiliki dua nilai yang dinyatakan dengan False dan True. Nilai ini bukan merupakan *string* atau *integer*, keduanya merupakan nilai spesial, hanya untuk variabel boolean. Tipe data boolean banyak dipakai untuk kode program percabangan atau untuk memutuskan apa yang mesti dijalankan ketika sebuah kondisi terjadi.

Sebagai contoh, untuk membuat kode program yang dapat menentukan apakah sebuah angka genap atau ganjil berdasarkan input dari pengguna, kita harus mengecek terlebih dahulu apakah angka tersebut habis dibagi 2 (untuk angka genap), atau tidak habis dibagi 2 (untuk angka ganjil). Tipe data boolean bisa dipakai untuk menampung kondisi seperti ini, yakni benar atau salah (True atau False). Penggunaan tipe data boolean ini akan lebih jelas saat kita masuk ke kondisi percabangan program seperti IF yang akan dibahas lebih lanjut nanti.

Cara Penggunaan Tipe Data Boolean Python

Sebagaimana yang sudah dijelaskan sebelumnya, tipe data boolean hanya bisa diisi dengan 2 nilai, yakni salah satu dari True atau False. Berikut contoh penulisannya:

```
foo = True
bar = False
print(foo)
print(bar)
```

Program tersebut, kita menginput nilai Boolean True kedalam variabel foo dan nilai Boolean False ke dalam variabel bar. Hasil kode program Python tersebut adalah:

```
True
False
```

Ada hal yang juga harus diperhatikan, yaitu penulisan huruf besar atau kecil. Dalam bahasa Python, penulisannya harus persis seperti itu, jika diinput sebagai true atau TRUE.

```
foo = true
print(foo)
```

Hasil kode program Python:

```
Traceback (most recent call last):
  File "D:\New folder (2)\latihan.py", line 1, in <module>
    foo = true
NameError: name 'true' is not defined
```

Terdapat *error* karena 'true' dianggap nama variable, bukan tipe data Boolean.

Selain diinput manual, tipe data boolean lebih sering didapat sebagai hasil dari operasi perbandingan, seperti apakah suatu angka lebih besar dari angka lainnya, apakah lebih kecil, atau sama dengan. Berikut contoh penggunaan operasi perbandingan ini:

```
foo = 12 < 10
print(foo)
foo = 12 > 10
print(foo)
foo = "A" == "a"
print(foo)
```

Hasil kode program Python:

```
False
True
False
```

- Di baris pertama, kita menyimpan hasil operasi perbandingan $12 < 10$ ke dalam variabel `foo`. Apakah $12 < 10$? Salah. Sehingga nilai variabel `foo` akan berisi boolean `False`.
- Di baris ketiga, operasi perbandingannya dibalik, yakni apakah $12 > 10$? Benar. Sehingga nilai variabel `foo` akan berisi boolean `True`.
- Terakhir di baris kelima operasi perbandingannya adalah apakah "A" sama dengan "a". Disini kita membandingkan antara dua buah *string*, namun karena "A" tidak sama dengan "a", maka variabel `foo` akan berisi boolean `False`.

Hasil boolean dari operasi perbandingan ini juga bisa didapat tanpa harus menyimpannya ke dalam variabel, seperti contoh berikut:

```
print(12 < 10)
print(12 > 10)
print("A" == "a")
```

Hasil kode program Python:

```
False
True
False
```

Operasi perbandingan yang dipakai masih sama seperti sebelumnya, hanya saja kali ini hasil perbandingan langsung diinput ke dalam perintah `print()`.

Lebih jauh lagi, metode perbandingan seperti ini sangat sering dipakai dalam struktur logika IF seperti contoh berikut:

BAB 5

PERULANGAN, TIPE DATA LIST, DAN OPERASI DALAM LIST

A. MATERI PEMBELAJARAN

1. Perulangan

Perulangan adalah baris kode atau instruksi yang dieksekusi oleh komputer secara berulang-ulang sampai suatu kondisi tertentu terpenuhi. Konsep perulangan ini didukung di semua bahasa pemrograman modern, termasuk Python.

Dengan perulangan, kita bisa mengeksekusi suatu kode program berkali-kali dengan jumlah tertentu, atau selama sebuah kondisi tertentu terpenuhi. Sebenarnya hampir sama seperti percabangan Python. Hanya saja bedanya:

- a. Pada percabangan, blok kode yang memenuhi kondisi tertentu hanya akan dieksekusi satu kali saja.
- b. Sedangkan perulangan akan dilakukan seterusnya berulang-ulang dengan jumlah tertentu atau selama kondisi tertentu terpenuhi.

Pada Python, kita bisa melakukan perulangan dengan beberapa cara di antaranya:

- a. Perulangan for
- b. Perulangan while
- c. Fungsi rekursif (akan dibahas pada Bab 6)

Perulangan For (Pada Python)

Perulangan for pada Python adalah perintah yang digunakan untuk melakukan iterasi dari sebuah nilai *sequence* atau data koleksi pada Python seperti List, Tuple, String dan lain-lain.

For pada Python memiliki perilaku yang berbeda dengan for pada kebanyakan bahasa pemrograman yang lain, karena for pada Python sangat berkaitan dengan data sequence atau data kolektif. Jika dibandingkan dengan bahasa lain, for pada Python lebih dikenal sebagai foreach. Berikut ini adalah struktur sintaks metode for:

```
for nilai in sequence:  
    # blok kode for
```

Jadi, ada 3 bagian penting, yaitu:

- a. sequence: adalah sebuah nilai yang bersifat *iterable* alias bisa diulang-ulang. Di antara tipe data yang bersifat *sequence* atau *iterable* adalah:
 - list
 - tuple
 - string
 - dan lain sebagainya
- b. nilai: adalah setiap *item* yang diekstrak dari sequence
- c. Blok kode: yaitu perintah-perintah yang akan dieksekusi secara berulang.

For dengan list

Perhatikan contoh berikut:

```
listNamaHewan = [  
    'Kucing', 'Kelinci', 'Ayam', 'Burung', 'Gajah',  
    'Jerapah', 'Domba', 'Sapi'  
]  
  
for NamaHewan in listNamaHewan:  
    print>NamaHewan)
```

Jika dieksekusi, program di atas akan menghasilkan output:

```
Kucing  
Kelinci  
Ayam  
Burung  
Gajah  
Jerapah  
Domba  
Sapi
```

Untuk mengetahui urutan iterasi for dengan list, kita bisa menggunakan fungsi enumerate. Fungsi tersebut akan mengekstrak 2 buah nilai:

- 1) yang pertama adalah index: yaitu urutan iterasi yang keberapa
- 2) dan *item* atau nilai dari list itu sendiri.

Perhatikan contoh berikut:

```
listNamaHewan = [  
    'Kucing', 'Kelinci', 'Ayam', 'Burung', 'Gajah',  
    'Jerapah', 'Domba', 'Sapi'  
]  
  
for i, NamaHewan in enumerate(listNamaHewan):  
    print(i, NamaHewan)
```

Kode program di atas sama saja seperti sebelumnya, kita hanya menambahkan fungsi enumerate() dan mem-*passing* variabel listNamaHewan sebagai parameter. Kita juga mengekstrak dua buah nilai yang kita kasih nama i dan NamaHewan. Jika dijalankan, berikut adalah output yang kita dapat:

```
0 Kucing  
1 Kelinci  
2 Ayam  
3 Burung  
4 Gajah  
5 Jerapah  
6 Domba  
7 Sapi
```

Perhatikan, urutannya dimulai dari **0**, bukan dari angka **1**.

Selain dengan list, kita juga bisa menggunakan for dengan fungsi range ().

Perhatikan contoh berikut:

```
## 0 sampai 4  
for i in range(5):  
    print("Perulangan ke -", i)
```

BAB 6

PENGGUNAAN

PEMROGRAMAN MODULAR

(FUNGSI)

A. MATERI PEMBELAJARAN

Fungsi pada Python adalah kumpulan perintah atau baris kode yang dikelompokkan menjadi satu kesatuan untuk kemudian bisa dipanggil atau digunakan berkali-kali. Sebuah fungsi bisa menerima parameter, mengembalikan suatu nilai, dan dipanggil berkali-kali secara independen. Dengan fungsi kita bisa memecah program besar yang kita tulis, menjadi bagian-bagian kecil dengan tugasnya masing-masing. Selain itu, fungsi akan membuat kode program kita menjadi lebih “*reusable*” dan lebih terstruktur.

1. Fungsi

Di dalam Python, sintaks pembuatan fungsi terlihat seperti berikut:

```
def <nama_fungsi>(parameters):  
    statements
```

Sintaks di atas secara umum terbagi menjadi 4 bagian:

1. Kata kunci `def` yang menjadi pertanda bahwa blok kode program adalah sebuah fungsi
2. Nama fungsi yang kita buat
3. Parameters yang akan diterima oleh fungsi yang kita buat (tidak wajib)
4. Dan blok kode fungsi yang berisi perintah-perintah yang harus dilakukan oleh sebuah fungsi. Blok kode program di dalam Python didefinisikan dengan indentasi.

Contoh sebuah fungsi sederhana dengan nama `halo_dunia()`:

```
def halo_dunia():  
    print('Halo python! Halo dunia!')
```

Ketika dijalankan, kode tersebut tidak akan menghasilkan apapun.

```
D:\>coba_python.py
```

```
D:\>
```

Mengapa begitu? Karena fungsi tersebut belum dipanggil. Sebuah fungsi akan dieksekusi jika dipanggil. Begitu pula fungsi `halo_dunia()` pun harus dipanggil terlebih dahulu supaya perintah `print('Halo Python! Halo dunia!')` dijalankan. Untuk memanggil fungsi yang telah kita definisikan, cukup ketik nama fungsinya, ditambah dengan tanda kurung `()` seperti berikut:

```
halo_dunia()
```

Output:

```
Halo python! Halo dunia!
```

Berkaitan dengan sifat fungsi yang *reusable*, kita bisa memanggil fungsi `halo_dunia()` berkali-kali:

```
halo_dunia()  
halo_dunia()  
halo_dunia()
```

Output:

```
Halo python! Halo dunia!  
Halo python! Halo dunia!  
Halo python! Halo dunia!
```

Fungsi dengan Argumen atau Parameter

Sebuah fungsi juga bisa menerima parameter atau pun argumen. Parameter merupakan suatu nilai/variabel yang dilemparkan ke dalam fungsi untuk diproses lebih lanjut. Sebagai contoh, perhatikan output berikut:

```
Halo Nalar, selamat datang!  
Halo Yusuf, selamat datang!  
Halo Hasya, selamat datang!  
Halo Syakira, selamat datang!
```

Bagaimana cara kita memproduksi output seperti itu dengan Python? Ada banyak cara. Bisa dengan list, perulangan, dan lain sebagainya. Namun, cara yang paling mudah adalah dengan melakukan 4x print() seperti ini:

```
print('Halo Nalar, selamat datang!')
print('Halo Yusuf, selamat datang!')
print('Halo Hasya, selamat datang!')
print('Halo Syakira, selamat datang!')
```

Akan tetapi, dari pada kita melakukan 4x print seperti di atas, kita bisa memanfaatkan fungsi dan parameter pada Python. Sehingga kode programnya akan terlihat seperti ini:

```
def selamat_datang (nama):
    print(f'Halo {nama}, selamat datang!')

selamat_datang('Nalar')
selamat_datang('Yusuf')
selamat_datang('Hasya')
selamat_datang('Syakira')
```

Dengan kode program tersebut, kita akan mendapatkan output yang sama dengan cara yang lebih elegan.

Parameter Wajib

Ada 2 jenis parameter di dalam Python, yaitu parameter wajib dan parameter opsional. Perhatikan contoh fungsi berikut:

```
def perkenalan (nama, asal):
    print(f'Perkenalkan saya {nama} dari {asal}')
```

Jika dipanggil:

```
perkenalan('Nalar Istiqomah', 'Jawa Barat')
```

Kita akan mendapatkan output:

```
Perkenalkan saya Nalar Istiqomah dari Jawa Barat
```

Tapi jika kita memanggilnya dengan parameter tidak lengkap, justru kita akan mendapatkan *error*:

```
perkenalan('Nalar Istiqomah')
```

BAB 7

PENGGUNAAN TUPLE,

DICTIONARY, DAN

PEMROSESAN DATA

A. MATERI PEMBELAJARAN

Pada bab 5, kita telah mempelajari salah satu tipe data kolektif pada Python, yaitu tipe data list. Sedikit mengulang, tipe data koleksi adalah jenis tipe data yang digunakan untuk menghimpun kumpulan data (atau data yang berjumlah lebih dari satu). Secara umum, terdapat 3 tipe data koleksi pada Python, yaitu:

1. List
2. Tuple
3. Dictionary

Masing-masing dari 3 tipe data di atas memiliki sifat dan kegunaan sendiri-sendiri. Agar kita tahu kapan kita membutuhkan tipe data a dan kapan kita membutuhkan tipe data b, maka kita harus mempelajari semuanya dengan baik. Dan pada pertemuan ini, kita akan coba berkenalan dengan tipe data Tuple dan Dictionary pada Python.

1. Tipe Data Tuple

Tuple adalah 1 dari 3 tipe data kolektif pada Python yang berguna untuk menyimpan lebih dari satu nilai dalam satu variabel secara sekaligus. Tuple bersifat *ordered* (terurut) dan juga bersifat *unchangeable* (tidak bisa diubah). *Ordered* berarti datanya bisa kita akses menggunakan indeks, dan *unchangeable* berarti datanya tidak akan pernah bisa diubah setelah pertama kali mendefinisikan. Dalam Python, tipe data tuple didefinisikan dengan tanda kurung ().

Tuple sama saja dengan list, sama-sama digunakan untuk menyimpan data himpunan. Sama-sama bisa menampung berbagai macam tipe data dalam satu himpunan. Hanya saja setelah diberi nilai, tuple tidak bisa diubah lagi. Selain itu, dari segi penulisan, list menggunakan kurung siku [] sedangkan tuple menggunakan kurung biasa ().

Cara membuat Tuple

Ada 3 cara untuk membuat tuple. Perhatikan contoh berikut:

```
# cara standar
tuple_jenis_kelamin = ('laki-laki', 'perempuan')

# tanpa kurung
tuple_status_perkawinan = 'menikah', 'lajang'

# menggunakan fungsi tuple()
tuple_lulus = tuple(['lulus', 'tidak lulus'])
```

Keterangan:

- Cara yang pertama adalah cara standar dan paling dasar.
- Cara yang kedua tanpa tanda kurung. Seperti memasukkan nilai ke variable biasa. Tapi pada tuple, nilai yang dimasukkan ada lebih dari 1.
- Cara yang ketiga adalah dengan menggunakan fungsi tuple() yang berfungsi mengubah list menjadi tuple

Tuple kosong

Untuk membuat tuple kosong, kita cukup dengan menuliskan dua tanda kurung seperti berikut:

```
tuple_kosong = ()
```

Tuple yang hanya berisi satu item

Untuk mendefinisikan tuple yang hanya berisi satu item, berbeda dengan list dengan elemen tunggal. Pada tuple, kita tetap diharuskan menulis tanda koma.

```
tuple_tunggal = (10,)
```

Jika tidak, maka Python akan menganggap tanda kurungnya tidak ada, seperti contoh berikut:

```
print(type((10))) # yang ini dianggap integer biasa
print(type((10,))) # yang ini dianggap tuple
```

Output:

```
<class 'int'>
<class 'tuple'>
```

Cara mengakses nilai tuple

Mengakses data pada tuple tidak jauh berbeda dengan cara mengakses data pada list, bahkan bisa dibilang sama persis. Hal ini disebabkan sifat list dan tuple yang *ordered*. Kita bisa mengakses nilai pada tuple dengan langsung mendefinisikan indeks-nya seperti berikut:

```
# cara standar
tuple_jenis_kelamin = ('laki-laki', 'perempuan')

print(tuple_jenis_kelamin[1]) # indeks satu
print(tuple_jenis_kelamin[0]) # indeks nol
```

Output:

```
perempuan
laki-laki
```

Kita juga bisa mengakses nilai pada tuple dengan negatif indeks:

```
# cara standar
tuple_jenis_kelamin = ('laki-laki', 'perempuan')

print(tuple_jenis_kelamin[-2])
print(tuple_jenis_kelamin[-1])
```

Output:

```
laki-laki
perempuan
```

BAB 8

BUDAYA DIGITAL

A. MATERI PEMBELAJARAN

Budaya digital adalah syarat penting kesuksesan sebuah transformasi digital, karena penerapan budaya digital adalah mengubah pola pikir (*mindset*) agar dapat beradaptasi dengan perkembangan digital. Budaya digital terdiri dari dua kata yaitu “Budaya” dan “Digital”. Kata “Budaya” berasal dari bahasa Sansakerta yaitu “buddhayah” yang artinya hal yang berhubungan dengan budi dan akal manusia. Budaya juga dapat diartikan sebagai suatu cara hidup yang dimiliki oleh sekelompok orang yang kemudian berkembang dan diwariskan dari generasi ke generasi. Dalam Bahasa Inggris, Budaya disebut dengan *culture*, yang artinya *the customs and beliefs, art, way of life and social organization of a particular country or group* (kebiasaan dan kepercayaan, seni, cara hidup dan organisasi sosial dari negara atau kelompok tertentu).

Adapun kata Digital berasal dari Bahasa Yunani yaitu Digitus, yang berarti jari jemari. Manusia memiliki 10 jari jemari, angka 10 itu terdiri dari angka 1 dan 0 yang merepresentasikan bilangan biner. Semua sistem komputer menggunakan sistem bilangan biner sebagai basis datanya.

Jadi, dapat disimpulkan bahwa budaya/kultur digital adalah konsep yang menggambarkan gagasan bahwa teknologi dan internet secara signifikan membentuk cara kita berinteraksi, berperilaku, berpikir, dan berkomunikasi sebagai manusia dalam lingkungan masyarakat. Terdapat kata kunci dari pengertian tersebut yaitu adanya penggunaan teknologi dan internet. Teknologi tersebut bisa kita gunakan dalam bentuk *smartphone*, laptop, komputer yang terhubung dengan jaringan internet.

Digital culture (budaya digital) juga dapat didefinisikan sebagai nilai atau *belief* dari pendekatan yang menyeluruh (*holistic*) dengan memanfaatkan teknologi digital untuk mengintegrasikan perilaku dan *mindset* dari pelaku

organisasi dalam upaya menangkap peluang dari pelanggan, menciptakan inovasi, kreativitas dan kolaborasi serta mitigasi resiko.

Salah satu contoh budaya digital dalam kehidupan sehari-hari adalah aktivitas di sosial media. Saat ini, banyak bermunculan *content creator* di sosial media (sosmed). Dengan melakukan *posting* di sosmed, para *content creator* mengekspresikan kontennya yang bermacam-macam, ada yang tentang *parenting*, resep masakan, komedi dan lainnya. Dari sini muncul istilah *digital content*. Dalam kesehariannya, *content creator* membuat konten dengan *gadget* (*smartphone*, *laptop* atau *tablet*) yang terhubung dengan jaringan internet. Ide-ide baru di konten yang dituangkan oleh *content creator*, selalu muncul setiap harinya. Sampai pada akhirnya kegiatan membuat konten menjadi sebuah kebiasaan dan bahkan kebutuhan. Masih banyak sekali contoh kultur digital, seperti belanja *online*, transportasi *online*, dan lainnya.

Mengapa Menanamkan Digital Culture Itu Penting?

Menanamkan *digital culture* kedalam organisasi atau perusahaan itu sangat penting. Karena transformasi digital tidak lengkap tanpa adanya budaya digital. Budaya sendiri terdiri dari nilai-nilai dan seperangkat karakteristik perilaku yang menentukan bagaimana hal-hal dilakukan dalam suatu organisasi. Budaya yang sehat memberikan pedoman/kode perilaku tidak tertulis yang mengarahkan individu untuk bertindak secara tepat dan membuat pilihan yang memajukan tujuan dan strategi organisasi. Terdapat tiga alasan penting untuk menanamkan *digital culture* selama transformasi digital, yaitu:

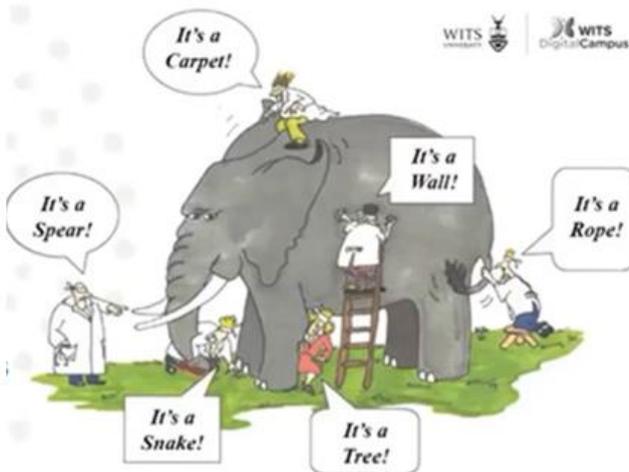
1. Transformasi digital yang dilakukan perusahaan beresiko mengalami kegagalan jika mengabaikan budaya digital. Hampir 80% perusahaan yang fokus pada penerapan budaya digital, bisa mempertahankan kinerjanya bahkan bisa memberikan terobosan baru.
2. Pergerakan organisasi atau perusahaan digital lebih cepat daripada organisasi tradisional. Oleh karena itu, pegawai atau anggota organisasi harus bisa cepat melakukan penilaian dan membuat keputusan setiap saat. Disinilah budaya digital diperlukan.
3. Generasi milenial saat ini banyak tertarik pada perusahaan digital karena *culture promise* mereka akan lingkungan kerja yang kreatif dan otonomi yang lebih besar. Situs web seperti [linkedin.com](https://www.linkedin.com) dan [glassdoor.com](https://www.glassdoor.com)

BAB 9

SYSTEM THINKING

A. MATERI PEMBELAJARAN

Systems thinking atau dalam bahasa Indonesia disebut dengan pemikiran sistemik adalah suatu pendekatan dalam memahami berbagai macam sistem dengan menekankan pada hubungan antar elemen yang ada pada sistem tersebut. Pada analisis tradisional, kita mempelajari suatu sistem dengan memisah-misahkan elemennya. Sedangkan dalam *system thinking*, kita melihat sistem dengan perspektif yang lebih luas. Hal ini menyebabkan output yang dihasilkan *system thinking* lebih akurat dan realistis. *System thinking* digunakan dalam penyelesaian masalah yang membutuhkan pemikiran holistik dan pemikiran reduksionis secara seimbang. Dengan memahami sistem secara keseluruhan dan detail, kita dapat menghindari kesalahan pada penafsiran masalah dan munculnya output atau solusi yang tidak diinginkan. Coba perhatikan gambar berikut:



Gambar 7 gambaran *system thinking*

Dapat dilihat bahwa ketika kita melihat sesuatu hanya sebagian, kita bisa salah menafsirkan. Contoh, ketika kita melihat gambar secara keseluruhan, kita tahu bahwa itu adalah seekor gajah. Tapi ketika hanya melihat sebagian, kita bisa menjadi salah dalam menafsirkan, ada yang bilang itu ular, pohon, tembok, dan lainnya.

Sebelum membahas *system thinking* lebih lanjut, kita harus mengenal terlebih dahulu apa itu “sistem”.

1. Sistem

Sistem merupakan sebuah konsep yang luas dan abstrak, sehingga seringkali kita mengalami kesulitan untuk menemukan dasar dari berbagai definisi tentang sistem. Dalam prakteknya, terminologi sistem telah sangat luas digunakan oleh berbagai disiplin ilmu, sehingga muncul berbagai pendapat dalam menafsirkannya. Berikut adalah beberapa definisi sistem:

- a. Sistem dapat digunakan untuk menunjuk suatu kumpulan yang disatukan oleh suatu bentuk saling ketergantungan, sehingga menjadi kesatuan yang bulat dan terpadu. Contohnya, sistem tata surya dan ekosistem.
- b. Sistem digunakan untuk menyebut alat-alat atau organ tubuh yang secara khusus memberikan andil terhadap berfungsinya tubuh tertentu. Contoh: sistem syaraf, sistem pencernaan.
- c. Sistem dapat digunakan untuk menunjuk sekumpulan gagasan yang tersusun, suatu himpunan gagasan, doktrin, prinsip, hukum dan sebagainya. Contoh: sistem pemerintahan demokratis, sistem nkri.
- d. Sistem dapat digunakan untuk menunjuk suatu hipotesis atau suatu teori. Contoh: pendidikan sistematis.
- e. Sistem juga menunjuk suatu cara atau metode. Contoh: sistem pembelajaran jarak jauh.
- f. Sistem dapat digunakan untuk menunjuk pengertian skema atau metode pengaturan organisasi atau susunan tertentu.

Namun, secara umum, sistem didefinisikan sebagai kumpulan bagian-bagian yang saling berinteraksi dan saling bergantung antar bagian-bagian pembentuknya untuk mempertahankan keberadaan dan fungsinya secara keseluruhan dalam rangka mencapai suatu tujuan di dalam lingkungan yang kompleks. Berdasarkan definisi tersebut, kita dapat mengidentifikasi lima poin penting yang berkaitan dengan sistem, yaitu:

BAB 10

CRITICAL THINKING

A. MATERI PEMBELAJARAN

Seseorang yang mempunyai banyak pengetahuan dan informasi belum tentu mempunyai *critical thinking*. Karena *critical thinking* bukan terbatas pada mengumpulkan banyak informasi, tapi juga mampu menyimpulkan inti dari informasi yang diketahuinya. Orang yang mempunyai *critical thinking* juga tahu bagaimana cara mencari sumber informasi yang relevan dan terpercaya serta memanfaatkan informasi tersebut untuk menyelesaikan masalahnya. Jadi, apa itu *critical thinking*? Berikut adalah pengertian *critical thinking* menurut beberapa ahli:

1. Glaser (2009), mendefinisikan *critical thinking skill* sebagai suatu sikap mau berpikir secara mendalam tentang masalah-masalah dan hal-hal yang berada dalam jangkauan pengalaman seseorang, pengetahuan tentang metode-metode pemeriksaan dan penalaran yang logis, dan suatu keterampilan untuk menerapkan metode-metode tersebut.
2. Yasushi Gotoh (2016) mendefinisikan “*Critical thinking as the set of skills and dispositions which enable one to solve problems logically and to attempt to reflect autonomously by means of Metacognitive regulation on one's own problem-solving processes.*”
3. Patricia C. Seifert (2010): “deciding what to do and when, where, why, and how to do it.”
4. Facione, Facione, and Sanchez (2010): “*Critical thinking is a process of making reasoned judgments based on the consideration of available evidence, contextual aspects of a situation, and pertinent concepts.*”
5. Menurut Ventista (2018) berpikir kritis adalah proses disiplin aktif dan aktif konseptualisasi, penerapan, analisis, sintesis, dan/atau mengevaluasi informasi yang dikumpulkan dari, atau dihasilkan melalui pengamatan, pengalaman, refleksi, penalaran, atau komunikasi secara terampil.

Berdasarkan pemaparan para ahli tersebut, dapat disimpulkan bahwa *critical thinking* adalah kemampuan untuk berpikir secara jernih dan rasional tentang apa yang harus dilakukan atau apa yang harus dipercaya. Seseorang dengan keterampilan *critical thinking* dapat melakukan hal-hal sebagai berikut:

1. Memahami hubungan logika antara gagasan
2. Mengidentifikasi, membangun, dan mengevaluasi argumen
3. Mendeteksi ketidak-konsistenan dan kesalahan umum dalam bernalar
4. Menyelesaikan masalah secara sistematis
5. Mengidentifikasi relevansi dan pentingnya gagasan
6. Merenungkan sebuah pembenaran keyakinan dan nilai-nilai dalam diri seseorang.

Berpikir kritis merupakan keterampilan berpikir universal yang berguna untuk semua profesi dan jenis pekerjaan. Selain itu, berpikir kritis bermanfaat dalam kegiatan sehari-hari seperti membaca, menulis, berbicara, mendengarkan, berdiskusi, dan sebagainya, untuk meningkatkan pemahaman terhadap suatu masalah. Pemikiran yang reflektif dan independen dapat menghindari keterikatan kepada keyakinan yang salah, sehingga memperkecil risiko untuk pengambilan keputusan salah.

Secara umum, *critical thinking* dibagi menjadi 3 (tiga) keterampilan, yaitu;

1. *Curiosity* atau keingintahuan, merupakan keinginan untuk mempelajari lebih banyak informasi-informasi dan mencari bukti-bukti, serta terbuka terhadap gagasan-gagasan baru.
2. *Skepticism* atau skeptis, merupakan sikap yang tidak mudah mempercayai semua informasi yang datang kepada kita. Sikap ini merupakan benteng akal sehat agar ketika informasi diterima, kita dapat membuat pertanyaan terlebih dahulu sebelum mengambil keputusan.
3. *Humility* atau kerendahan hati, merupakan kemampuan kita untuk mengakui bahwa pendapat dan gagasan kita adalah salah saat ditampilkan bukti-bukti baru yang membuktikan bahwa gagasan kita memang salah.

Berikut adalah contoh critical thinking:

| Tingkatan/Jenis Keterampilan Berpikir Kritis | Contoh Keterampilan Berpikir Kritis |
|--|---|
| Mendefinisikan dan mengklarifikasi masalah | <ol style="list-style-type: none"> 1. Mengidentifikasi inti dari isu atau masalah. 2. Mengkomparasi persamaan-persamaan dan perbedaan-perbedaan. 3. Menentukan manakah informasi yang relevan. 4. Memformulasi pertanyaan-pertanyaan dengan tepat. |
| Menentukan informasi-informasi yang relevan dengan masalah | <ol style="list-style-type: none"> 1. Membedakan antara fakta, opini, dan keputusan logis. 2. Mengecek konsistensi. 3. Mengenali stereotip dan klise. 4. Mengenali bias, faktor-faktor emosional, propaganda, dan istilah semantik. 5. Mengenali nilai sistem dan ideologi yang berbeda. |
| Menyelesaikan masalah / menggambarkan konklusi | <ol style="list-style-type: none"> 1. Mengenali ketepatan data. 2. Memprediksi kemungkinan-kemungkinan konsekuensi |

1. Manfaat Critical Thinking

Kata “kritis” seringkali diartikan sebagai tindakan atau pemikiran argumentatif terhadap orang lain. Padahal, berpikir kritis tidak sama dengan berdebat atau mengkritisi orang lain. Justru, *critical thinking* mempunyai banyak manfaat baik bagi diri sendiri maupun orang sekitar kita. Manfaat dari *critical thinking* antara lain:

a. Aset yang tidak akan pernah habis

Apapun profesi kita baik itu di bidang pendidikan, hukum, manajemen, keuangan dan lainnya, *critical thinking* penting untuk dimiliki. Kemampuan untuk berpikir jernih dan rasional dengan *critical thinking* merupakan aset penting untuk karir apapun.

BAB 11

IoT: ARSITEKTUR, PROTOKOL, DAN TEKNOLOGI, SERTA PENERAPANNYA

A. MATERI PEMBELAJARAN

Internet of Things atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet. *Internet of Things* (IoT) adalah sebuah konsep/skenario dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. "*A Things*" pada *Internet of Things* dapat didefinisikan sebagai subjek, misalkan orang dengan monitor *implant* jantung, hewan peternakan dengan *transponder biochip*, dan sebuah mobil yang telah dilengkapi *built-in* sensor untuk memperingatkan pengemudi ketika tekanan ban rendah.

1. Sejarah Internet of Things

Alat internet pertama yang dikembangkan dengan konsep IoT adalah mesin Coke (Coca Cola) di Carnegie Mellon University pada tahun 1989. Para *programmer* dapat terhubung ke mesin melalui Internet, sehingga mereka bisa memeriksa apakah ada minuman atau tidak di mesin tanpa harus mendatangi mesin tersebut. Kemudian pada tahun 1993, peneliti di Cambridge University mengembangkan webcam online pertama. Awalnya, mereka harus berjalan ke mesin pembuat kopi untuk melihat apakah kopi sudah terisi atau belum. Mereka lalu memasang kamera di dekat mesin pembuat kopi yang akan mengirimkan foto setiap 3 menit sekali, sehingga mereka tidak perlu jalan bolak balik mengecek mesin kopi.

Istilah IoT (Internet of Things) mulai dikenal tahun 1999 yang saat itu disebutkan pertama kalinya dalam sebuah presentasi oleh Kevin Ashton, cofounder and executive director of the Auto-ID Center di MIT, yang dikenal sebagai “Father of IoT”.

Lebih lanjut pada tahun 2000, LG Electronics membuat Digital DIOS Refrigerator atau lebih dikenal dengan kulkas pintar. Kulkas ini dapat digunakan untuk *browsing* internet, mencari resep atau menonton tutorial, bahkan bisa melakukan video call. Samsung juga mulai merambah dan mengembangkan teknologi IOT di bidang consumer appliances seperti : Smart Air Conditioner, Smart TV, Smart Refrigerator. Menurut CEO Samsung, pada tahun 2017, 90% dari semua produk Samsung berupa perangkat IOT, termasuk semua televisi dan perangkat mobile. Kompetitor terdekat Samsung, yakni Apple pun memiliki upaya di bidang IOT dengan proyek HomeKit, yang merupakan protokol pengontrol rumah pintar melalui sistem operasi iOS. Beberapa produk Apple tersebut antara lain iHome, Incipio, GridConnect, dan iDevices.

PBB memprediksi bahwa IoT akan menjadi *the next big thing*. Oleh karena itu, pada tahun 2007 Apple merilis iPhone pertama yang langsung menjadi pusat perhatian dunia. Walaupun *smartphone* sudah lama ada sebelumnya, iPhone menjadi branding tersendiri karena menggunakan internet untuk beroperasi. Sampai pada tahun 2008, pengguna IoT semakin banyak. Bahkan menurut Cisco IBSG, jumlah perangkat IoT lebih banyak daripada jumlah populasi saat itu.

Selanjutnya, industri IoT semakin berkembang dalam skala besar. Banyaknya penerapan IoT ini menghasilkan pembangunan infrastruktur dengan konsep *smart city*. Pada tahun 2014, Seoul menjadi *smart city* pertama, kemudian disusul oleh Singapura, Amsterdam dan New York.

Menginjak tahun 2015, sensor banyak diterapkan pada *smartphone*. Karena sensor inilah, penggunaan IoT banyak ditemukan pada perangkat yang bisa dipakai (*wearable*) seperti *smartwatch*, alat pelacak GPS, dan alat kesehatan. Contohnya: Google Glass, Google Nest, Nike Fit, dan Samsung Smart Watch.

IoT terus berkembang dan dipakai dalam banyak bidang. Menurut laporan terbaru Juniper Research, jumlah perangkat IoT pada tahun 2021 mencapai 46 miliar. Sebagai perbandingan, ini meningkat 200%

BAB 12

KECERDASAN BUATAN DAN PENERAPANNYA

A. MATERI PEMBELAJARAN

Berikut adalah definisi kecerdasan buatan atau *artificial intelligence* (AI), definisinya menurut beberapa pakar:

1. Schalkoff (1990): AI adalah bidang studi yang berusaha menerangkan dan meniru perilaku cerdas dalam bentuk proses komputasi.
2. Rich dan Knight (1991): AI adalah studi tentang cara membuat komputer melakukan sesuatu yang, sampai saat ini, orang dapat melakukannya lebih baik.
3. Luger dan Stubblefield (1993): AI adalah cabang ilmu komputer yang berhubungan dengan otomasi perilaku yang cerdas.
4. Haag dan Keen (1996): AI adalah bidang studi yang berhubungan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia dalam sebuah sistem teknologi informasi sehingga sistem tersebut dapat memfasilitasi proses pengambilan keputusan yang biasanya dilakukan oleh manusia.

Berdasarkan pemaparan para ahli tersebut, dapat disimpulkan bahwa kecerdasan buatan atau *artificial intelligence* (AI) merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan didunia ini karena manusia mempunyai pengetahuan dan pengalaman yang diperoleh dari proses belajar. Semakin banyak bekal pengetahuan yang dimiliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan banyak pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Sebaliknya, dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

Jika dibandingkan dengan kecerdasan alami (kecerdasan yang dimiliki oleh manusia), kecerdasan buatan memiliki beberapa kelebihan, yaitu:

1. Kecerdasan buatan bersifat permanen.
Karena sifat alami manusia yaitu lupa, kecerdasan alami dimungkinkan akan mengalami perubahan. Sedangkan kecerdasan buatan tidak akan berubah sepanjang sistem komputer & komputer tidak mengubahnya.
2. Kecerdasan buatan lebih mudah diduplikasi & disebar.
Mentransfer pengetahuan manusia dari orang ke orang lain membutuhkan proses yang sangat lama (melalui proses Pendidikan). Selain itu, suatu keahlian itu tidak akan pernah bisa diduplikasi dengan lengkap. Adapun pengetahuan yang terletak pada pada suatu sistem komputer, dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.
3. Kecerdasan buatan lebih murah dibanding dengan kecerdasan alami.
Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Kecerdasan buatan bersifat konsisten.
Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah (karena sifat manusia yang mudah lupa dan tidak konsisten).

BAB 13

MACHINE LEARNING DAN PENERAPANNYA

A. MATERI PEMBELAJARAN

Machine Learning (ML) merupakan bidang studi yang fokus pada desain dan analisis algoritme sehingga memungkinkan komputer bisa belajar. ML merupakan algoritme yang bersifat umum (generik) dimana komputer dapat belajar tanpa deprogram secara eksplisit. Program tersebut memanfaatkan data untuk membangun model dan mengambil keputusan berdasarkan model yang telah dibangun. Mitchel (1997) mendefinisikan bahwa ML adalah “program komputer yang dapat melakukan pembelajaran dari pengalaman E (*Experience*) terhadap tugas T (*Task*) dan mengukur peningkatan kinerja P (*Performance Measure*), jika kinerja Tugas T diukur oleh kinerja P, maka meningkatkan pengalaman E”. Dari definisi ini, dapat dikatakan sebuah aplikasi *Machine Learning* memiliki 3 komponen yaitu *Task* T, *Performance Measure* P, dan *Experience* E. Oleh karena itu, untuk membangun sebuah aplikasi ML maka komponen T, P dan E harus dapat diidentifikasi.

Task T merupakan tujuan dari program ML yang kita buat. Program ML membantu kita untuk mengatasi masalah yang sulit untuk diselesaikan oleh program tradisional yang bersifat statis. Berikut ini beberapa contoh Task yang ada program ML:

1. Klasifikasi adalah suatu pengelompokan data dimana data yang digunakan tersebut mempunyai kelas label atau target. Pada jenis task ini, program komputer diminta untuk “menebak” atau memilih kelompok/kategori/kelas dari data. Contohnya adalah klasifikasi emosi pada foto manusia dengan tiga kelas yaitu senang, marah dan netral.

Aplikasi tersebut harus memilih salah satu (dalam kasus tertentu bisa lebih dari satu) dari tiga kelas tersebut berdasarkan foto yang diberikan.

2. Regresi sederhananya adalah sebuah tugas dimana aplikasi ML diminta untuk menebak angka (bilangan kontinu) berdasarkan sejumlah data. Sebagai contoh kita ingin melakukan prediksi harga saham, maka luaran dari prediksi tersebut adalah sebuah nilai harga. Jadi perbedaan antara regresi dan klasifikasi adalah objek yang diprediksi, regresi memprediksi angka sedangkan klasifikasi memprediksi kelas.
3. Transkripsi adalah sebuah tipe *Task* dimana aplikasi ML mencoba “memahami” sebuah data yang tidak terstruktur yang merepresentasikan data lain. Sebagai contoh aplikasi text to speech dimana data berupa gelombang suara. Proses mengubah gelombang suara menjadi teks adalah transkripsi.
4. *Machine Translation*. *Task* ini merupakan proses mengubah sekumpulan input terurut yang berisi simbol-simbol tertentu menjadi simbol tertentu dibahasa lainnya. Contoh aplikasinya penerjemah bahasa Indonesia ke bahasa Inggris.
5. *Anomaly Detection*. *Task* ini bertujuan untuk menggunakan aplikasi ML untuk memeriksa sejumlah data atau *event* lalu menandainya sebagai sesuatu yang tidak biasa.
6. *Syntesis* dan *Sampling*. *Task* ini bertujuan untuk membuat atau *generate* sesuatu berdasarkan contoh. Misalnya aplikasi ML yang dapat menghasilkan sebuah gambar pemandangan berdasarkan sketsa tertentu.

Dalam membangun model *machine learning*, terdapat dua tahapan yaitu *training* dan *testing*. *Training* adalah proses konstruksi model dan *testing* adalah proses menguji kinerja model pembelajaran. Kedua tahapan ini menggunakan dataset untuk membuat dan mengevaluasi model. Dataset nantinya akan dibagi menjadi tiga jenis yang tidak beririsan (suatu sampel pada himpunan tertentu, tidak muncul pada himpunan lainnya):

1. *Training set* adalah himpunan data yang digunakan untuk melatih atau membangun model.
2. *Development set* atau *validation set* adalah himpunan data yang digunakan untuk mengoptimasi saat melatih model. Model dilatih menggunakan training set dan pada umumnya kinerja saat latihan diuji dengan validation

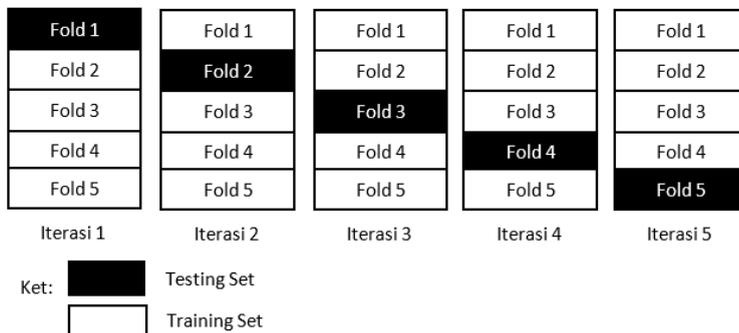
set. Hal ini berguna untuk generalisasi (agar model mampu mengenali pola secara generik).

3. *Testing set* adalah himpunan data yang digunakan untuk menguji model setelah proses pelatihan selesai.

Perlu diperhatikan bahwa *training*, *validation* dan *testing* data secara ideal diambil (*sampled*) dari distribusi yang sama dan memiliki karakteristik yang sama (*independently and identically distributed*). Distribusi pada masing-masing dataset ini juga sebaiknya seimbang (*balanced*) dan memuat seluruh kasus. Misal, sebuah dataset *binary classification* (dua kelas) sebaiknya memuat 50% kasus positif dan 50% kasus negatif.

Pada umumnya, rasio pembagian data set (*training*, *validation* dan *testing*) adalah 80%, 10% dan 10% atau 90%, 5% dan 5%. Namun, pada umumnya, *validation set* tidak digunakan apabila dataset berukuran kecil. Untuk dataset berukuran kecil biasanya hanya dibagi *training* dan *testing set* saja. Dalam kasus ini, pembagian dataset menjadi *training* dan *testing set* pada umumnya memiliki rasio (90%, 10%), (80%, 20%), (70%,30%) atau (50%,50%).

Saat tidak menggunakan *validation set*, kita juga memiliki opsi untuk mengevaluasi model dengan metode *K-cross-validation*. Artinya, kita membagi *training* dataset menjadi *K* bagian. Kita menggunakan *K-1* bagian untuk *training*, kemudian menguji kinerja model saat latihan (*validation*) menggunakan satu bagian. Hal ini diulang sebanyak *K* kali dimana sebuah bagian data digunakan sebagai *testing set* sebanyak sekali (bergilir).



Gambar 28 Contoh K-fold validation dengan k=5

BAB 14

BIG DATA

A. MATERI PEMBELAJARAN

Big data, atau disebut juga dengan data raya, maha data, data raksasa atau data badang adalah istilah umum untuk menggambarkan data dalam jumlah yang sangat besar, rumit, dan tak terstruktur sehingga menjadikannya sulit diolah dengan hanya menggunakan perkakas manajemen pangkalan data biasa atau aplikasi pengolah data tradisional.

Menurut McKinsey Global (2011), *big data* dapat didefinisikan dengan data yang memiliki skala (*volume*), distribusi (*velocity*), keragaman (*variety*) yang sangat besar, dan atau abadi, sehingga membutuhkan penggunaan arsitektur teknikal dan metode analitik yang inovatif untuk mendapatkan wawasan yang dapat memberikan nilai bisnis baru (informasi yang bermakna). Adapun menurut Hurwitz (2013), *big data* merupakan istilah untuk sekumpulan data yang begitu besar atau kompleks dimana tidak bisa ditangani lagi dengan sistem teknologi komputer konvensional.

Jadi, kapan suatu data dapat dikatakan sebagai “Big Data”? Apa yang membedakan *big data* dengan data konvensional? Berikut adalah karakteristik data yang termasuk *big data*:

1. *Volume*

Jumlah data yang dihasilkan sangat besar, berukuran petabyte (10^{15}) sampai zettabyte (10^{21}). Contohnya:

- a. Facebook menghasilkan 10TB data baru setiap hari, Twitter 7TB
- b. Sebuah Boeing 737 menghasilkan 240 terabyte data penerbangan selama penerbangan dari satu wilayah bagian AS ke wilayah yang lain
- c. Google, Microsoft dan Amazon kini memiliki satu juta server untuk menampung data-datanya.

2. *Velocity*

Velocity adalah kecepatan data yang masuk (per jam, per detik, atau *real time*). Selain ukurannya yang besar, *big data* juga ditandai dengan kecepatan pertambahannya yang tinggi. Contoh, data *clickstreams* (*web log*) dan transfer data *asynchronous* pada web yang dapat menangkap apa saja aktifitas yang dilakukan oleh jutaan atau lebih *user* web secara bersamaan. Sebuah *clickstream* merekam setiap *gesture* yang dibuat oleh pengunjung dan *gesture* ini memiliki potensi untuk memberikan deskripsi mengenai kebiasaan dari pengunjung. Data *clickstream* ini akan mengidentifikasi sesi yang berhasil dan tidak berhasil, menentukan apakah pengunjung puas atau tidak puas, dan menemukan bagian dari *website* yang secara efektif menarik perhatian pengunjung (misal, untuk menentukan daerah mana yang efektif untuk memasang iklan).

3. *Variety*

Variety merupakan kumpulan dari berbagai macam data, baik data yang terstruktur, semi terstruktur maupun data tidak terstruktur (pada *big data*, paling banyak data yang tidak terstruktur). Tampilan data semakin komprehensif (lengkap dan menyeluruh).

Dan pada pengembangannya ada yang menyebut karakteristik *big data* itu 5V termasuk Volume, Velocity, Variety, Variability, Veracity. Ada juga yang menyebut 7V bahkan 10V.

1. Perkembangan *Big Data* (BD)

Berikut adalah perkembangan *big data* dari tahun ke tahun

***Big Data* 1.0 (1994-2004)**

Big data 1.0 mengacu pada era teknologi web yang digunakan untuk menganalisis aktifitas *online* pengguna. Analisis web dapat dibagi tiga yaitu penggunaan web, struktur web, dan konten web. Analisis penggunaan web (*web usage*) adalah analisis dengan menggunakan aplikasi web untuk memahami perilaku pengguna web selama melakukan *browsing* (perilaku). Dengan cara ini, pihak pengelola atau pembuat web dapat menyiapkan layanan web yang sesuai dengan pengguna.

Analisis struktur web adalah proses analisis struktur web atau *web page*. Struktur web ini terdiri dari *web page* sebagai *nodes* dan *hyperlink* yang menghubungkan antara *page*. *Hyperlink* menghubungkan lokasi web

page ke lokasi web page lainnya, baik di dalam web yang sama atau berbeda. Google's PageRank menganalisis struktur *hyperlink web page* untuk meranking mereka berdasarkan popularitas atau pentingnya *pages*. Analisis *content web* adalah proses pemanfaatan informasi dari konten web. *Web content mining* berhubungan dengan ekstraksi informasi *web page*, *clustering web page*, klasifikasi *web page*. Contohnya untuk identifikasi *cyberterrorism*, *email fraud*, *spam mail filtering*.

Big Data 2.0 (2005-2014)

BD 2.0 mengacu pada perkembangan *social media*, dimana pada era ini, *user* web dapat berinteraksi langsung dengan pemilik konten web, bahkan *user* web dapat memberikan kontribusi data dan informasi untuk konten web (misal, *posting* status atau foto). Pada big data 2.0 ini, banyak dilakukan *social media analisis* (SMA) untuk menganalisis dan menginterpretasikan perilaku manusia pada *social media*, mengkaji perkembangan pengetahuan, memahami kepentingan *user*, cara-cara *browsing* web, *friend list*, *sentimen*, *profession*, dan *opinion*. SMA digunakan untuk menganalisis bahasa yang natural (bahasa sehari-hari), dan content yang independen. SMA focus pada *Sentiment Analysis* (SA), dan *Social Network Analysis* (SNA).

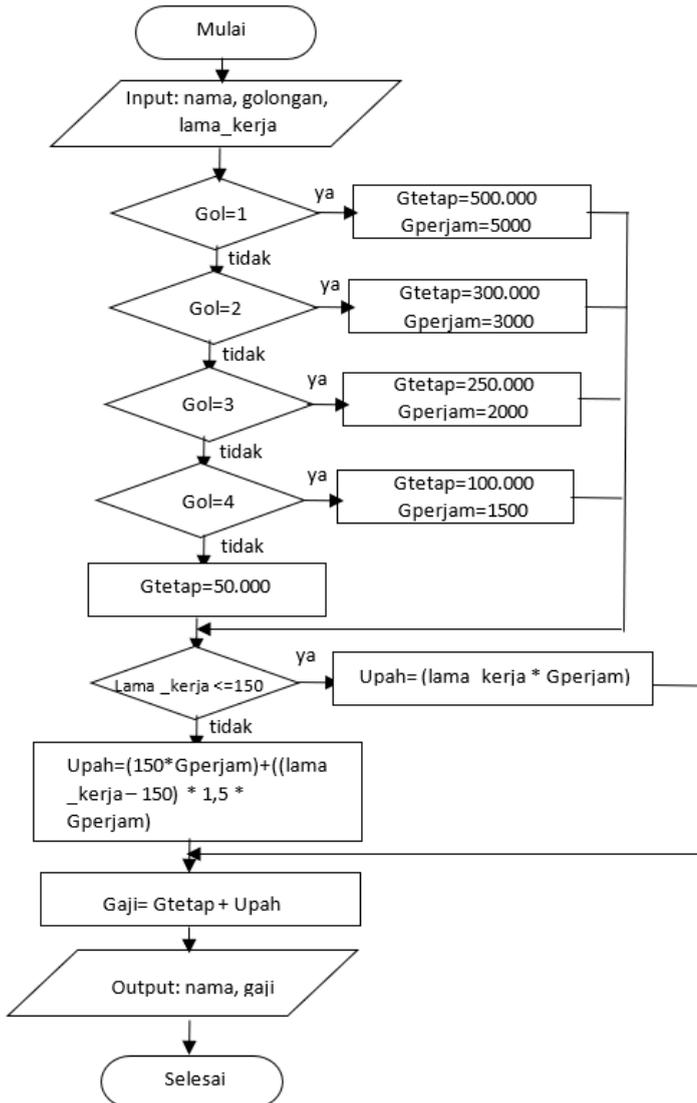
Sentiment Analysis (SA) menggunakan analisis teks, proses bahasa yang natural (*natural language processing*), dan bahasa komputer untuk mengidentifikasi sentimen atau opini pengguna dari sebuah teks. SA dapat digunakan untuk *multi level* analisis mulai dari *entity level*, kalimat, dan dokumen. *Entity level analysis* mengidentifikasi opini *user* yang ada dalam dokumen. Analisis kalimat mengidentifikasi poin yang ditekankan pada kalimat. Analisis dokumen mengidentifikasi hal sentimen yang difokuskan pada dokumen. SA digunakan untuk menganalisis pergerakan *stock market*, memahamai *trend market*, analisis produk, dan manajemen krisis. Pada umumnya SA menggunakan *tools of application program interface* (API) untuk menganalisis data yang diperlukan. Berikut ini list *tools of application program interface* (API)

- a. Google Maps API
- b. You Tube APIs
- c. Flickr API

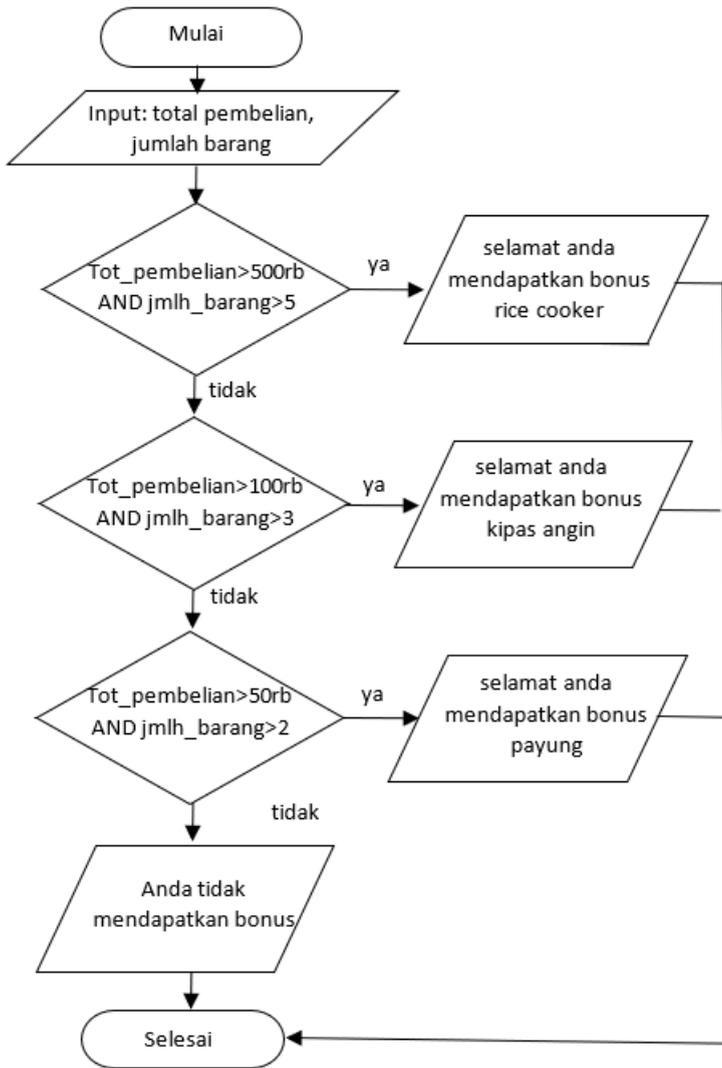
KUNCI JAWABAN

BAB 2

5. Flowchartnya adalah sebagai berikut:



6. Flowchartnya adalah sebagai berikut:



| | | |
|----|---|---|
| 36 | 6 | $\sqrt{(23 - 36)^2 + (5 - 6)^2} = 13,038$ |
| 36 | 5 | $\sqrt{(23 - 36)^2 + (5 - 5)^2} = 13$ |
| 41 | 3 | $\sqrt{(23 - 41)^2 + (5 - 3)^2} = 18,111$ |
| 32 | 4 | $\sqrt{(23 - 32)^2 + (5 - 4)^2} = 9,055$ |
| 32 | 2 | $\sqrt{(23 - 32)^2 + (5 - 2)^2} = 9,487$ |

- c. Urutkan jarak tersebut dan tetapkan tetangga terdekat berdasarkan jarak minimum ke-K

| Umur | Pendapatan | Jarak Euclid dengan (23,5) | Peringkat | 3 Tetangga terdekat? |
|------|------------|----------------------------|-----------|----------------------|
| 42 | 6 | 19,026 | 6 | Tidak |
| 36 | 6 | 13,038 | 4 | Tidak |
| 36 | 5 | 13 | 3 | Ya |
| 41 | 3 | 18,111 | 5 | Tidak |
| 32 | 4 | 9,055 | 1 | Ya |
| 32 | 2 | 9,487 | 2 | Ya |

- d. Periksa kelas dari tetangga terdekat

| Umur | Pendapatan | Peringkat | Kelas Tetangga terdekat |
|------|------------|-----------|-------------------------|
| 36 | 5 | 3 | Lancar |
| 32 | 4 | 2 | Lancar |
| 32 | 2 | 1 | Tidak Lancar |

- e. Gunakan mayoritas sederhana dari kelas tetangga terdekat sebagai nilai prediksi data baru

Hasil pada langkah d menunjukkan bahwa dari 3 tetangga terdekat, terdapat 2 kelas Lancar dan 1 kelas Tidak Lancar, maka disimpulkan bahwa data baru (23,5) termasuk ke dalam kelas Lancar.

DAFTAR PUSTAKA

- Cholissodin I., Riyandani, E. 2018. Analisis Big Data (Teori & Aplikasi) “Big Data vs Big Information vs Big Knowledge”. Fakultas Ilmu Komputer Universitas Brawijaya
- Han Jiawei, Kamber Micheline. 2006. Data Mining: Concepts and Techniques Second Edition. Morgan Kaufmann Publishers.
- Santoso, JT. 2022. Proyek Coding dengan Python. Yayasan Prima Agus Teknik
- Shina A.F.I, dkk. 2021. INDONESIA CAKAP DIGITAL Pendekatan Integrasi-Interkoneksi Keislaman. Penerbit Samudra Biru.
- Trilestari E.W, Almamalik L. 2018. *Systems Thinking: Suatu Pendekatan Pemecahan Permasalahan yang Kompleks dan Dinamis*. STIA LAN Bandung Press.

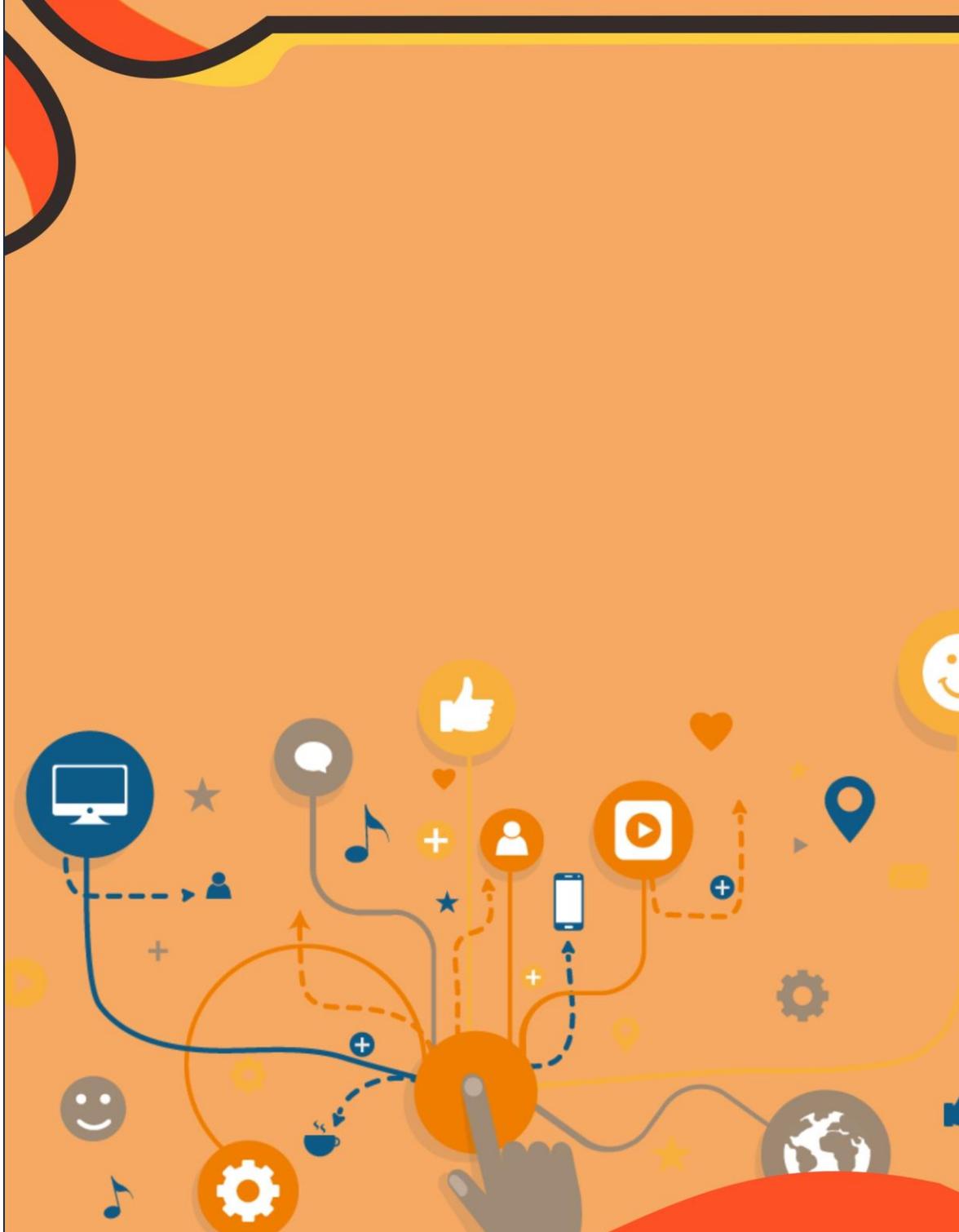
BIODATA PENULIS

Nalar Istiqomah, MKom lahir pada tanggal 23 Agustus 1993 di Subang. Penulis merupakan lulusan Ilmu Komputer Institut Pertanian Bogor. Semasa kuliah S1, penulis mengikuti program *fast track* dan berhasil lulus S1-S2 selama 5 tahun. Penulis mengambil bidang minat Data Mining dan Statistik. Setelah lulus S1, penulis bekerja sebagai *Research Fellow* di Divisi Statistika dan Divisi Kebijakan Makro Prudensial Bank Indonesia. Selain itu, penulis juga pernah bekerja sebagai Divisi Marketing di PT. Karya Muda Inspira. Saat ini, penulis merupakan dosen di Sekolah Tinggi Manajemen Aktuaria Trisakti (STMA Trisakti).

SINOPSIS

Transformasi Digital merupakan pengetahuan dasar yang melatih *system thinking*, *critical thinking* dan *computational thinking* untuk membangun cara-cara baru dalam berbagai aktivitas manusia dan/atau menyelesaikan persoalan yang dihadapi. Modul ini membahas masalah-masalah khas era digital, bentuk-bentuk transformasi dan disrupsi yang terjadi pada masa kini serta melatih *system thinking*, *critical thinking* dan *computational thinking* untuk memberikan gagasan solusi dari masalah tersebut. Pada modul ini, *computational thinking* dilatih dengan proses pengembangan program komputer secara sederhana menggunakan bahasa pemrograman Python. Selain itu, dikenalkan pula teknologi yang ada pada saat ini seperti IoT, *Artificial Intelligence* (AI) atau Kecerdasan Buatan, *Machine Learning* serta *Big Data*.

Hal-hal tersebut merupakan pengetahuan dasar yang dibutuhkan untuk menghadapi era revolusi industri 4.0. Dimana pada masa ini, perkembangan teknologi sangat pesat sehingga kita perlu mengetahui dan mengidentifikasi teknologi mana yang bisa digunakan untuk masalah tertentu. Pada masa ini, ukuran data pun tidak lagi ratusan, atau ribuan, tapi bisa jadi ratusan ribu bahkan tak terhingga (*Big Data*). Oleh karena itu sangat dibutuhkan kemampuan pemrograman untuk mengolah data mulai dari praproses data, kemudian mengolahnya untuk *forecasting*, klasifikasi, *clustering*, dan lain sebagainya. Setelah mempelajari modul ini, diharapkan kita dapat merelasikan teknologi TIK sebagai *tools* dalam *project* yang dikerjakannya, meskipun tidak sampai pada tahapan implementasi (*coding*).



CV. Tahta Media Group
Surakarta, Jawa Tengah
Web : www.tahtamedia.com
Ig : tahtamediagroup
Telp/WA : +62 896-5427-3996

ISBN 978-623-147-262-5 (PDF)



9 786231 472625