

Editor

Yusuf Wahyu Setiya Putra, M.Kom.
Hanson Prihantoro Putro, S.T., M.T., CTFL.



ALGORITMA PEMROGRAMAN



Wika Purbasari, Taufiq Iqbal, Inayatul Inayah,
Munawir, Sri Sutjiningtyas, Erna Hikmawati, Fauzan Natsir,
Anak Agung Sandatya Widhiyanti, Muhammad Wali,
M. Syauqi Haris, Hasan Basri

Editor

Yusuf Wahyu Setiya Putra, M.Kom.
Hanson Prihantoro Putro, S.T., M.T., CTFL.

ALGORITMA PEMROGRAMAN

Wika Purbasari, Taufiq Iqbal, Inayatul Inayah,
Munawir, Sri Sutjiningtyas, Erna Hikmawati, Fauzan Natsir,
Anak Agung Sandatya Widhiyanti, Muhammad Wali,
M. Syauqi Haris, Hasan Basri



Haura Utama

KATA PENGANTAR

Dengan penuh rasa syukur, kami menghadirkan buku "Algoritma Pemrograman" ke hadapan pembaca. Rasa syukur ini kami persembahkan kepada Tuhan Yang Maha Esa, yang telah memberikan kesempatan, ilmu, dan kekuatan sehingga buku ini dapat terwujud. Buku ini merupakan hasil dari perjalanan panjang dalam memahami dan mengajarkan algoritma, yang merupakan dasar penting dalam dunia informatika dan ilmu komputer.

Buku ini dirancang untuk memberikan pemahaman yang mendalam tentang algoritma, mulai dari konsep dasarnya, pengembangan algoritma dalam bentuk *flowchart* dan *pseudocode*, hingga implementasinya dalam bahasa pemrograman C++. Setiap bab telah disusun untuk memastikan bahwa pembaca mendapatkan pemahaman yang lengkap, tidak hanya dari segi teori tapi juga praktik. Buku ini mengeksplorasi berbagai topik seperti Tipe Data, Perulangan, Penyeleksian, Prosedur, Fungsi, Array, Record, hingga konsep lebih lanjut seperti Rekursi serta Algoritma Pencarian dan Pengurutan, semuanya disajikan dengan bahasa yang mudah dipahami dan aplikatif. Kami berharap buku ini dapat menjadi panduan yang berharga bagi pelajar, mahasiswa, dan siapa saja yang ingin memperdalam pengetahuan mereka dalam algoritma pemrograman.

Terakhir, kami juga ingin mengucapkan terima kasih kepada semua pihak yang telah mendukung proses penulisan buku ini. Semoga buku ini dapat memberikan manfaat yang luas dan menjadi bagian dari perjalanan pembelajaran kita semua.

Bandung, Januari 2024

Penulis

Algoritma Pemrograman, karya Wika Purbasari, M.Kom, dkk,
diterbitkan pertama kali oleh Penerbit Haura Utama, 2024

18.2 x 25.7 cm, vi + 242 hlm

Hak cipta dilindungi undang-undang
Dilarang mereproduksi atau memperbanyak seluruh
maupun sebagian dari buku ini dalam bentuk dan
cara apapun tanpa izin tertulis dari penerbit

Editor : Yusuf Wahyu Setiya Putra, M.Kom.
Hanson Prihantoro Putro, S.T., M.T., CTFL.
Penata isi : Zulfa
Perancang sampul : Nita



CV. Haura Utama

Anggota IKAPI Nomor 375/JBA/2020
Nagrak, Benteng, Warudoyong, Sukabumi
+62877-8193-0045 haurautama@gmail.com

Cetakan I, Januari 2024

ISBN: 978-623-492-783-2

Reference Book

 penerbithaura.com

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
1. ALGORITMA DAN FLOWCHART	1
1.1. Pengantar	1
1.2. Algoritma Deskriptif	7
1.3. <i>Pseudocode</i>	9
1.4. Desain <i>Flowchart</i>	10
1.5. Pengantar Bahasa Pemrograman.....	17
2. TIPE DATA.....	20
2.1. Pengantar	20
2.2. Definisi Tipe Data	21
2.3. Jenis Tipe Data.....	24
2.4. Implementasi Tipe Data	28
2.5. Penutup	38
3. STRUKTUR PERULANGAN.....	40
3.1. Pengantar	40
3.2. Definisi Perulangan	41
3.3. Jenis Struktur Perulangan dan Implementasinya	43
3.4. Latihan Soal.....	63
3.5. Penutup	63
4. STRUKTUR PENYELEKSIAN	64
4.1. Pengantar	64
4.2. Definisi Penyeleksian.....	66
4.3. Jenis Struktur Penyeleksian	67
4.4. Contoh Implementasi Penyeleksian Kasus #1	69
4.5. Prinsip Dasar Struktur Penyeleksian	72
4.6. Contoh Implementasi Penyeleksian Kasus #2	74

4.7. Penutup	79
5. PROCEDURE	82
5.1. Pengantar	82
5.2. Tujuan dan Manfaat Prosedur	82
5.3. Penggunaan Prosedur	83
5.4. Parameter	87
5.5. Studi Kasus	98
5.6. Latihan Soal.....	115
6. FUNCTION	117
6.1. Pengantar	117
6.2. Tujuan dan Manfaat Fungsi.....	118
6.3. Fungsi tanpa Parameter	120
6.4. Fungsi dengan Parameter	124
6.5. Fungsi Bersarang.....	126
7. ARRAY	129
7.1. Pengantar	129
7.2. Deklarasi dan Inisialisasi Array	133
7.3. Dimensi Array	139
7.4. Penutup	145
8. RECORD	148
8.1. Pengantar	148
8.2. Dasar Teori.....	149
8.3. Akses Elemen <i>Record</i>	151
8.4. Record Bersarang (<i>Nested Record</i>)	154
8.5. <i>Record</i> dengan Array	157
8.6. Latihan Soal.....	162
8.7. Penutup	165
9. REKURSI	166
9.1. Pengantar	166
9.2. Deklarasi dan Penggunaan	168
9.3. Prinsip Struktur Rekursi	172

9.4. Kasus dan Implementasi Rekursi	174
9.5. Rekursi Tanpa Basis.....	182
9.6. Penutup	185
10. ALGORITMA PENCARIAN	187
10.1. Pengantar	187
10.2. <i>Linear Search</i>	188
10.3. <i>Binary Search</i>	191
10.4. <i>Interpolation Search</i>	194
11. ALGORITMA PENGURUTAN.....	197
11.1. Pengantar	197
11.2. <i>Bubble Sort</i>	199
11.3. <i>Selection Sort</i>	204
11.4. <i>Insertion Sort</i>	211
11.5. <i>Merge Sort</i>	218
11.6. Penutup	227
DAFTAR PUSTAKA	228
BIODATA PENULIS	236

1. ALGORITMA DAN FLOWCHART

1.1. Pengantar

Algoritma diambil dari kata **Al-Kwarizmi** yang kemudian berubah menjadi algorism, yang selanjutnya menjadi algorithm, dan berdasarkan catatan sejarah pencetus algoritma pertama kali adalah seorang ahli matematika dan astronomi Bernama **Abu Ja'far Muhammad Ibnu Musa Al-Kwarijmi** yang berasal dari Persia, hal ini dibuktikan dalam buku yang di tulisnya di mana terdapat penjelasan langkah dalam menyelesaikan berbagai persoalan aritmatika (780-850 masehi). Algoritma merupakan ilmu pengetahuan dasar yang sangat penting bagi seorang *programmer* dalam menyelesaikan suatu permasalahan secara rinci dan terstruktur dengan cara menganalisa permasalahan terlebih dahulu kemudian menyusunnya ke dalam algoritma, kemudian algoritma yang sudah disusun dituangkan ke dalam bentuk program, setelah itu program yang sudah jadi di eksekusi dan diuji untuk bisa menghasilkan *output* dari permasalahan yang dihadapi. Dari penjelasan tersebut terdapat beberapa definisi dari algoritma yaitu (Dr. Suarga, 2012):

- (1) Algoritma merupakan kalimat dengan jumlah kata terbatas yang tersusun secara logis dan sistematis.

2. TIPE DATA

2.1. Pengantar

Dalam pemrograman, tipe data merupakan konsep dasar yang digunakan untuk menentukan jenis data yang dapat disimpan dan diolah dalam sebuah program. Pemilihan tipe data yang tepat memiliki dampak signifikan terhadap performa dan akurasi algoritma (Adomavicius & Zhang, J., 2012; Dogan & Tanrikulu, Z., 2013).

Dalam bab ini, Anda akan menggali lebih dalam tentang tipe data, sebuah konsep yang mungkin terlihat sederhana pada pandangan pertama, namun memiliki implikasi yang sangat luas dan mendalam dalam pemrograman komputer. Pemahaman terkait tipe data merupakan fondasi dari semua instruksi dan operasi dalam pemrograman. Ini adalah cara bagaimana program mengenali dan memproses berbagai bentuk data - dari angka dan teks hingga keputusan logis dan struktur data yang lebih kompleks. Tipe data tidak hanya tentang penyimpanan; ini adalah tentang representasi dan interpretasi nilai-nilai dalam sistem komputasi. Setiap tipe data memiliki aturan dan karakteristik tertentu yang menentukan bagaimana data tersebut disimpan, diakses, dan dimanipulasi.

Selanjutnya, Anda akan mengeksplorasi berbagai aspek tipe data dalam pemrograman. Mulai dari tipe data primitif hingga tipe

data kompleks, kita akan melihat bagaimana setiap tipe data berperan dalam pembuatan program yang efisien, efektif, dan aman. Kita juga akan membahas tentang bagaimana perubahan dalam teknologi komputasi mempengaruhi evolusi tipe data dan tantangan yang muncul bersamanya. Akhirnya, kita akan menyelami bagaimana pemahaman yang mendalam tentang tipe data sangat penting dalam praktek pemrograman yang baik dan pengembangan perangkat lunak yang sukses.

2.2. Definisi Tipe Data

Tipe data adalah klasifikasi yang menentukan jenis nilai yang dapat disimpan dan dioperasikan oleh suatu variabel. Dalam pemrograman, tipe data bukan hanya soal bagaimana data disimpan, tetapi juga bagaimana data tersebut diinterpretasikan dan diproses oleh komputer. Setiap tipe data memiliki ciri khas, termasuk ukuran memori yang digunakan dan cara komputer melakukan operasi pada data tersebut. Pemilihan tipe data yang tidak tepat dapat menyebabkan masalah seperti penggunaan memori yang tidak efisien, kesalahan dalam perhitungan, dan kerentanan dalam keamanan program.

Tipe data primitif seperti integer, float, double, char, dan boolean adalah konsep dasar yang ditemukan dalam hampir semua bahasa pemrograman (Sebesta, 2012). Mereka membentuk blok

3. STRUKTUR PERULANGAN

3.1. Pengantar

Pernahkah anda atau teman anda sewaktu duduk di bangku Sekolah Dasar melakukan kesalahan dan dihukum oleh guru untuk menulis di buku catatan “Saya menyesal dan tidak akan mengulanginya lagi” sebanyak 100 kali atau lebih? Jika kalimat tersebut ditulis secara manual, tentunya membutuhkan waktu yang cukup lama dan tenaga ekstra. Dalam bahasa pemrograman, kalimat tersebut dapat dituliskan secara otomatis menggunakan perintah-perintah khusus.

Bab ini akan mempelajari bagaimana kita dapat menulis kalimat tersebut tanpa menuliskan sebanyak 100 kali. Kita dapat menuliskan perintah tersebut menggunakan pengkodean dalam bahasa pemrograman untuk menjalankan perintah secara berulang-ulang dengan lebih efektif dan sederhana, baik perintah dengan pengkondisian perulangan yang telah ditentukan maupun pengkondisian perulangan yang bergantung pada suatu aturan tertentu. Struktur perulangan dapat diimplementasikan pada kondisi yang memerlukan iterasi atau perulangan secara sistematis, sehingga dapat meningkatkan efisiensi, menghemat waktu, meningkatkan produktivitas, serta mengurangi kesalahan.

3.2. Definisi Perulangan

Perulangan atau *looping* dalam algoritma pemrograman adalah perintah atau instruksi kode program yang bertujuan untuk mengulang beberapa baris kode program atau sekumpulan perintah selama kondisi masih terpenuhi. Dengan menggunakan perulangan, proses berulang dapat ditulis dengan beberapa baris koding saja, sehingga dapat meningkatkan efisiensi dan memperpendek kode program. Proses perulangan biasanya digunakan untuk mengulang proses pemasukan data, mengulang proses penghitungan, dan mengulang proses menampilkan hasil pengolahan data.

Secara umum perulangan dibedakan menjadi dua jenis, yaitu:

1. *Counted loop*: perulangan yang jumlah perulangannya sudah diketahui terlebih dahulu.
2. *Uncounted loop*: perulangan yang jumlah perulangannya belum diketahui kapan akan berhenti.

Penggunaan jenis perulangan disesuaikan dengan jenis kasusnya, sehingga memerlukan pemahaman dan analisis karakteristik penggunaan pada setiap jenis perulangan beserta operator-operator yang dapat digunakan dalam proses penyelesaian kasus yang akan diselesaikan.

4. STRUKTUR PENYELEKSIAN

4.1. Pengantar

Di era digital yang terus berkembang, algoritma pemrograman dan struktur penyeleksian khususnya, telah menjadi aspek penting dalam pengembangan perangkat lunak dan aplikasi. Struktur penyeleksian dalam pemrograman adalah sebuah konstruksi yang memungkinkan program untuk mengambil keputusan dan melakukan aksi yang berbeda tergantung pada kondisi yang diberikan. Bab ini akan menggali lebih dalam mengenai konsep ini, memperlihatkan bagaimana struktur penyeleksian tidak hanya fundamental dalam pembuatan algoritma yang efisien, tetapi juga krusial dalam mengembangkan solusi pemrograman yang fleksibel dan tangguh.

Struktur penyeleksian menyediakan kerangka kerja bagi programmer untuk mengatur alur eksekusi program. Dengan menggunakan struktur penyeleksian, programmer dapat menentukan kondisi tertentu yang, jika terpenuhi, akan menyebabkan blok kode tertentu dieksekusi. Konsep ini penting karena tidak semua kode dalam sebuah program harus dijalankan dalam setiap situasi. Misalnya, dalam aplikasi web, tindakan tertentu mungkin hanya dilakukan jika pengguna telah masuk ke akun mereka. Dalam kasus

ini, struktur penyeleksian memungkinkan program untuk memilih jalur eksekusi yang benar.

Pentingnya struktur penyeleksian dalam pemrograman telah diakui dan diteliti secara luas. Sebuah studi oleh Singhal dan Kumar (2000) mendemonstrasikan bagaimana struktur penyeleksian bersarang dapat digunakan untuk meningkatkan pemecahan masalah dalam pemrograman C/C++. Dalam pengujian dan pemeliharaan perangkat lunak, Rothermel, Harrold, dan Dedhia (2000) meneliti tentang pemilihan tes regresi untuk perangkat lunak C++. Studi ini, mengeksplorasi bagaimana struktur penyeleksian memainkan peran dalam optimasi proses pengujian perangkat lunak, khususnya dalam bahasa pemrograman C++.

Pembahasan tentang struktur penyeleksian tidak lengkap tanpa mempertimbangkan kaitannya dengan struktur data dan algoritma. Goodrich, Tamassia, dan Mount (2011) dalam buku mereka "*Data Structures and Algorithms in C++*" memberikan gambaran mendalam tentang bagaimana struktur penyeleksian diterapkan dalam algoritma dan struktur data. Buku ini menjadi referensi penting bagi mereka yang ingin memahami aplikasi praktis dari konsep ini dalam pemrograman C++. Selain itu, Oualline (2003) dalam bukunya "*Practical C++ Programming*" juga membahas tentang penggunaan praktis dari struktur penyeleksian dalam konteks pemrograman C++. Buku ini menawarkan perspektif yang berguna tentang bagaimana

5. PROCEDURE

5.1. Pengantar

Procedure atau prosedur adalah subprogram dalam pemrograman yang terdiri dari serangkaian pernyataan atau langkah-langkah yang didefinisikan sebagai satu kesatuan untuk mengeksekusi tugas tertentu. Prosedur juga disebut modul, subrutin (*sub-routine*), function atau method, tergantung pada bahasa pemrograman yang digunakan. Setiap istilah ini memiliki makna dan peran yang sedikit berbeda, tetapi semuanya merujuk pada konsep yang sama, yaitu memecah program menjadi bagian-bagian yang lebih kecil dan terstruktur.

Dengan pendekatan ini, pemrogram dapat mengembangkan program yang lebih mudah dipahami, dikelola, dan disusun. Prosedur dapat digunakan kembali dalam berbagai bagian program atau dalam program lain, memungkinkan efisiensi dalam pengembangan perangkat lunak. Sebuah prosedur memiliki nama yang digunakan untuk memanggilnya dan sering menerima parameter sebagai input.

5.2. Tujuan dan Manfaat Prosedur

Proses yang melibatkan memecah program besar menjadi modul-modul yang lebih kecil disebut sebagai modularisasi. Ada tiga alasan utama untuk melakukan modularisasi:

1. Modularitas: Dengan memecah kode menjadi prosedur-prosedur yang lebih kecil, kode menjadi lebih mudah dikelola, dibaca, dan diperbaiki. Setiap prosedur bertanggung jawab atas tugas tertentu.
2. Abstraksi: Modularisasi memberikan tingkat abstraksi dalam pemrograman, yang berarti pemrogram dapat fokus pada detail-detail penting di dalam setiap modul tanpa perlu memahami seluruh program secara keseluruhan. Ini membuat kode lebih mudah dipahami dan dikelola.
3. Kerja Kolaboratif: Modularisasi memungkinkan beberapa pemrogram untuk bekerja pada masalah yang sama secara bersamaan. Setiap pemrogram dapat fokus pada pengembangan atau pemeliharaan modul tertentu, sehingga memungkinkan proyek dikerjakan lebih cepat dan lebih efisien.
4. Reusabilitas: Dengan modularisasi, kode yang telah dikembangkan dalam satu proyek dapat digunakan kembali dalam proyek-proyek lain. Ini menghemat waktu dan upaya dalam pengembangan perangkat lunak, karena modul-modul yang telah ada dapat diintegrasikan kembali ke dalam proyek-proyek baru.

5.3. Penggunaan Prosedur

5.3.1. Mendefinisikan Prosedur

Prosedur dapat memiliki parameter ataupun tanpa parameter yang memungkinkan membuat sebuah prosedur yang dapat menerima satu atau lebih argumen sebagai input dan mengembalikan

2. Buatlah sebuah prosedur yang menerima tiga parameter bilangan bulat dan menentukan nilai maksimum di antara ketiganya. Tampilkan nilai maksimum tersebut.
3. Buatlah prosedur yang menerima sebuah array bilangan bulat dan parameter integer (x). Proses dari prosedur ini adalah menampilkan pesan "x ditemukan di dalam array" jika nilai x terdapat dalam array, atau "x tidak ditemukan di dalam array" jika tidak.
4. Misal array1 berisikan {3, 2, 1, 8, 5, 13, 21} dan array2 berisikan {1, 5, 3, 7, 13}. Buatlah prosedur yang akan menyalin nilai yang ada pada array1 dan array2 ke array3. Array1 dan array2 mungkin tidak terurut dan panjang array1 tidak selalu sama dengan array2.

6. FUNCTION

6.1. Pengantar

Function atau fungsi merupakan salah satu hal yang penting untuk dipahami dalam pembelajaran algoritma. Function adalah sub program yang terpisah dari program utama, mirip seperti prosedur yang telah dibahas pada bab sebelumnya. Perbedaan utama function dan procedure yaitu pada function bukan hanya berupa sekumpulan program yang terpisah tetapi dapat mengembalikan sebuah nilai untuk pemanggilnya (Indahyanti and Rahmawati, 2020).

Function merupakan salah satu bagian dari konsep awal pemrograman yaitu paradigma atau pendekatan fungsional. Dalam paradigma pemrograman fungsional, fungsi digunakan untuk membuat kode lebih terorganisir. Dengan membagi kode menjadi fungsi yang lebih kecil, setiap bagian dari program dapat dipahami dengan lebih mudah. Fungsi digunakan juga untuk menghindari pengulangan atau redundansi, mempermudah pengujian. Selain itu, fungsi juga mendukung konsep pure functions dan immutability. Dalam pemrograman fungsional, fungsi sering kali tidak memiliki efek samping dan tidak mengubah state program.

Namun demikian, penggunaan fungsi tetap diperlukan dalam paradigma pemrograman prosedural. Penggunaan fungsi membantu

Dapat kita lihat pada baris ke-7, di dalam function `volumebalok` terdapat pemanggilan function `luasPersegi` itulah yang disebut dengan function bersarang.

7. ARRAY

7.1. Pengantar

Array adalah kumpulan item dengan tipe variabel yang sama yang disimpan di lokasi memori yang berdekatan. Ini adalah salah satu struktur data paling populer dan sederhana dan sering digunakan untuk mengimplementasikan struktur data lainnya. Setiap item dalam array diindeks dimulai dengan 0, sehingga array disebut sebagai struktur data statis dengan ukuran tetap dan menggabungkan data dengan tipe serupa (Zailani, Apriyanto dan Zakaria, 2020). Array ini akan mengumpulkan item yang disimpan di lokasi memori yang berdekatan. Idennya adalah untuk menyimpan beberapa item dengan tipe yang sama secara bersamaan. Hal ini memudahkan penghitungan posisi setiap elemen hanya dengan menambahkan offset ke nilai dasar, yaitu lokasi memori elemen pertama array (umumnya dilambangkan dengan nama array).

Untuk mempermudah pemahaman, kita dapat membayangkan array pada Gambar 7.1 sebagai sebagai sebuah tangga yang di setiap langkahnya ditempatkan sebuah nilai. Di sini, akan didapatkan dalam mengidentifikasi lokasi hanya dengan mengetahui jumlah langkah yang dijalani. Hal ini memudahkan penghitungan posisi setiap elemen hanya dengan menambahkan offset ke nilai dasar, yaitu lokasi memori

8. RECORD

8.1. Pengantar

Dalam dunia pemrograman, struktur data memegang peran krusial dalam penyimpanan dan pengorganisasian informasi. *Record* atau disebut juga *structure*, sebagai salah satu bentuk struktur data, memungkinkan para pengembang untuk mengelompokkan data dengan karakteristik serupa ke dalam satu entitas. *Record* memberikan fleksibilitas yang luar biasa dalam merepresentasikan entitas-entitas kompleks dengan atribut-atribut yang berbeda.

Bab ini akan mengajak pembaca dalam perjalanan eksplorasi yang menyeluruh terhadap konsep *record*. Mulai dari pengenalan dasar hingga penerapan praktisnya dalam pemrograman. Selain itu, pembaca akan menemukan penjelasan tentang bagaimana *record* bekerja, bagaimana *record* dapat digunakan dalam memodelkan data, dan bagaimana implementasi *record* untuk memperkaya kemampuan pemrograman. Contoh dan latihan yang disertakan pada bab ini bertujuan untuk memberikan pemahaman yang mendalam dan aplikatif.

Pengetahuan tentang *record* menjadi fundamental dalam mengelola informasi yang kompleks dalam pemrograman. Dengan membaca bab ini, pembaca diharapkan akan memiliki pemahaman

tentang *record* sehingga memberikan dasar yang kuat bagi pembaca yang ingin mengembangkan dalam merancang solusi yang efisien dan aplikatif dalam pemrograman. Hal ini juga memungkinkan pembaca untuk memahami konsep - konsep yang lebih kompleks dalam rekayasa perangkat lunak.

8.2. Dasar Teori

Sama seperti array, *record/struct* memiliki sejumlah bagian yang disebut *field*. Sedangkan semua elemen array harus memiliki jenis data yang serupa, setiap elemen pada *record/struct* dapat memiliki variasi jenis data yang berbeda (Wilis Kaswidjanti, S.Si., n.d.). Sebuah *record* atau struktur dinyatakan dengan memanfaatkan *keyword* yaitu 'struct' atau dalam penulisannya pada bahasa C++ dikenal dengan nama struct (I Komang Setia Buana, Gusti Ngurah Mega Nata, 2018). Berikut adalah pseudocode untuk mendefinisikan sebuah *record*.

```
BEGIN
  STRUCT
    DECLARE name: STRING
    DECLARE age: INTEGER
    DECLARE address: STRING
  Person
END STRUCT
END
```

Pseudocode di atas mendeskripsikan sebuah struktur bernama *Person* yang memiliki tiga anggota: *name* dengan tipe data string, *age*

9. REKURSI

9.1. Pengantar

Pada bab ini, Anda akan memasuki dunia rekursi dalam pemrograman. Bagian ini adalah kelanjutan dari pemahaman tentang dasar-dasar pemrograman yang telah dipelajari dalam bab-bab sebelumnya, seperti Algoritma dan Flowchart (lihat bab 1), Tipe Data (lihat bab 2), Struktur Perulangan (lihat bab 3), Struktur Penyeleksian (lihat bab 4), Prosedur (lihat bab 5), Fungsi (lihat bab 6), Array (lihat bab 7), dan Record (lihat bab 8). Sekarang, Anda akan menjelajahi topik yang lebih kompleks tetapi sangat penting dalam pengembangan perangkat lunak, yaitu rekursi. Pada Bab 9 akan membahas topik penting dalam pemrograman yang dikenal sebagai rekursi. Rekursi adalah teknik pemrograman yang kuat yang memungkinkan sebuah fungsi untuk memanggil dirinya sendiri. Ini mungkin terdengar kontradiktif pada awalnya, tetapi rekursi memungkinkan penyelesaian masalah yang lebih besar dipecahkan menjadi masalah yang lebih kecil, yang kemudian dipecahkan kembali sampai masalah dasar teratasi. Dengan kata lain, kita dapat memecahkan masalah besar menjadi potongan-potongan kecil yang lebih mudah dikelola.

Rekursi adalah konsep yang mendasari banyak algoritma dalam ilmu komputer. Secara sederhana, rekursi adalah kemampuan sebuah fungsi untuk memecah masalah menjadi sub-masalah yang

lebih kecil. Fungsi tersebut kemudian memanggil dirinya sendiri untuk menyelesaikan sub-masalah tersebut. Prinsip ini mencerminkan cara yang seringkali memecahkan masalah dalam kehidupan sehari-hari, dengan membaginya menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola. Ini adalah ciri khas dari rekursi - memecah masalah menjadi versi yang lebih sederhana dari masalah itu sendiri. Rekursi memiliki daya ungkit yang kuat dalam mengekspresikan ide-ide dan algoritma yang kompleks. Ia memberikan cara berpikir yang berbeda dalam menghadapi masalah, dengan melihatnya dari sudut pandang yang lebih abstrak.

Anda mungkin bertanya-tanya mengapa kita perlu menggunakan rekursi ketika telah memiliki konsep perulangan (*looping*) untuk mengulangi tugas berulang kali. Jawabannya terletak pada jenis masalah yang dapat diselesaikan dengan rekursi. Rekursi sangat efektif ketika memiliki masalah yang memiliki struktur hierarki atau berulang yang dalam. Salah satu contoh klasik yang sering disebutkan adalah perhitungan faktorial. Faktorial dari sebuah angka adalah hasil dari perkalian semua bilangan bulat positif mulai dari 1 hingga angka tersebut. Faktorial sering kali dinyatakan sebagai $n!$ dengan n adalah angka yang ingin dihitung faktorialnya. Misalnya, $5!$ (baca: lima faktorial) adalah $5 \times 4 \times 3 \times 2 \times 1 = 120$. Dalam hal ini, Anda dapat merumuskan perhitungan faktorial sebagai masalah rekursif. Faktorial dari n adalah n dikali faktorial dari $(n-1)$. Namun, juga harus memikirkan basis kasus, yaitu faktorial dari 0 adalah 1. Dengan kata

proyek-proyek pemrograman, meningkatkan efisiensi, dan menghasilkan solusi yang lebih efektif.

Langkah selanjutnya Anda akan mempelajari bab-bab berikutnya. Pada Bab 10, Anda akan mendalami topik Algoritma Pencarian, yang akan membantu dalam mencari elemen-elemen dalam kumpulan data dengan berbagai metode pencarian (lihat bab 10). Kemudian, pada Bab 11, Anda akan menjelajahi topik Algoritma Pengurutan, yang memberikan beragam alat dan teknik untuk mengurutkan data dengan efisien (lihat bab 11). Dengan dasar yang kuat dalam rekursi dan pemahaman yang baik tentang algoritma-algoritma dasar ini, Anda akan semakin siap untuk menghadapi tantangan pemrograman yang lebih kompleks dan membangun solusi-solusi yang lebih canggih. Teruslah berlatih, eksperimen dengan konsep-konsep ini, dan Anda akan menjadi seorang pemrogram yang semakin mahir, kreatif, dan percaya diri dalam menghadapi berbagai masalah pemrograman yang menantang.

10. ALGORITMA PENCARIAN

10.1. Pengantar

Pencarian (*searching*) adalah hal yang sering kita lakukan dalam kehidupan sehari-hari. Kita seringkali melakukan pencarian sebuah kata tertentu dalam sebuah berkas naskah untuk menggantinya dengan kata yang lain. Dalam konteks ilmu komputer, pengambilan dan organisasi data merepresentasikan prinsip dasar algoritma. Proses iteratif terhadap elemen dalam struktur data untuk ekstraksi informasi spesifik dikenal sebagai pencarian. Sebaliknya, penyortiran merujuk pada proses penataan elemen dalam struktur data berdasarkan kriteria tertentu. Terdapat beragam algoritma pencarian dan penyortiran yang dikhususkan untuk jenis struktur data tertentu. Bab ini mengkhususkan diri pada teknik pencarian sehingga Anda akan memiliki pemahaman mendalam tentang penerapan algoritma pencarian yang paling sering digunakan.

Proses pencarian merupakan aktivitas yang bertujuan untuk menemukan elemen spesifik dalam sebuah struktur data. Ketika berurusan dengan *array*, ada dua metode utama yang dipilih berdasarkan status pengurutan *array* tersebut. Bab ini akan membahas tentang metode penelusuran linier dan biner. Penelusuran linier terkenal akan kemampuannya yang universal, mampu diterapkan


```

        LOC = I;
        // kasus khusus jika Key ada pada posisi I
    }
    return LOC;
    // Return lokasi KEY jika ditemukan atau -1 jika
    tidak
}

int main() {
    int A[] = {12, 13, 16, 18, 21, 22, 33, 35}; //
    contoh array terurut
    int N = sizeof(A) / sizeof(A[0]); // jumlah elemen
    pada array A
    int KEY = 18; // contoh nilai yang dicari
    int location = PencarianInterpolasi(A, N, KEY);
    if (location != -1) {
        std::cout << "Key found at index: " << location
        << std::endl;
    } else {
        std::cout << "Key not found in the array." <<
        std::endl;
    }
    return 0;
}

```

11. ALGORITMA PENGURUTAN

11.1. Pengantar

Pengurutan (*sorting*) adalah suatu proses di mana elemen-elemen dalam sebuah array atau list diatur ulang dengan tujuan untuk menyusun data dalam urutan tertentu. Urutan ini dapat berupa urutan menaik (*ascending*) yaitu elemen-elemen disusun dari nilai terkecil ke nilai terbesar, atau urutan menurun (*descending*) yaitu elemen-elemen disusun dari nilai terbesar ke nilai terkecil.

Tujuannya utama dari pengurutan adalah untuk mempermudah pencarian data dan menganalisis data dengan lebih efisien dan dapat membantu meningkatkan kinerja aplikasi komputasi. Pengurutan merupakan salah satu aspek dasar dari pemrograman dan komputasi yang sering kali diabaikan, namun memiliki peran yang sangat penting dalam berbagai aplikasi komputasi.

Salah satu pertanyaan yang sering muncul adalah kenapa sebuah data harus diurutkan ?, jawaban yang paling sederhana yaitu agar data dapat dengan mudah dilihat. Contoh yang mudah, ketika kita mendapatkan data tentang penjualan pada sebuah toko yang tidak terurut, yang berisi nama produk dan frekuensi penjualannya dalam satu bulan. Bayangkan bagaimana kita dapat mengetahui dengan cepat produk mana saja yang sangat laku dalam satu bulan ?.

DAFTAR PUSTAKA

- Adomavicius, G. and Zhang, J., 2012. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems (TMIS)*, 3(1), pp.1-17.
- Afrizal Zein, & Emi Sita Eriana. (2022). *Algoritma dan Struktur Data* (Emi Sita Eriana, Ed.; 1st ed.). Unpam Press.
- Akmal, S.Si., M., & Mira Suryani, S.Pd., M. K. (2018). *MODUL PRAKTIKUM ALGORITMA DAN PEMROGRAMAN*.
- Bae S. 2019. *JavaScript Data Structures and Algorithms*. Apress
- Bae, S. (2019) *JavaScript Data Structures and Algorithms: An Introduction to Understanding and Implementing Core Data Structure and Algorithm*. Jerman: Apress.
- Carey, J., Doshi, S., Rajan, P., 2019. *C++ Data Structures and Algorithm Design Principles: Leverage the Power of Modern C++ to Build Robust and Scalable Applications*. United Kingdom: Packt Publishing.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2002). *Introduction to Algorithms*. The MIT Press.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2022. *Introduction to Algorithms*, Fourth Edition. United Kingdom: MIT Press.
- Cortadella, J., Gavalda, R. and Orejas, F., *Introduction to Programming (in C++) Recursion*.
- Cowan, C., Wagle, F., Pu, C., Beattie, S. and Walpole, J., 2000, January. Buffer overflows: Attacks and defenses for the vulnerability of the decade. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00* (Vol. 2, pp. 119-129). IEEE.

- Dale, N. and Wheems, C. (2010) *Programming and Problem Solving with C++: Brief Edition*. Fifth Edit. Amerika Serikat: Jones & Bartlett Learning.
- Dale, N.B. and Weems, C., 2014. *Programming and problem solving with C++*. Jones & Bartlett Publishers.
- Dianta, I.A., 2021. *Logika dan Algoritma Untuk Merancang Aplikasi Komputer*. Yayasan Prima Agus Teknik, Semarang.
- Dogan, N. and Tanrikulu, Z., 2013. A comparative analysis of classification algorithms in data mining for accuracy, speed and robustness. *Information Technology and Management*, 14, pp.105-124.
- Donovan, A., Kernighan, B., 2015. *The Go Programming Language*. United Kingdom: Pearson Education.
- Dr. Suarga, M. Sc. , M. Math. , Ph. D. (2012). *Algoritma dan Pemrograman* (Sigit Suyantoro, Ed.; 2nd ed.). C.V Andi Offset.
- Drozdek, A., 2012. *Data Structures and algorithms in C++*. Cengage Learning.
- Elenia, E. E. *et al.* (2020) "Modul Praktikum Struktur Data," *Akuntansi Keuangan Lanjut 2*, (38), hal. 10.
- Elshaw, R., Sakr, S., Talia, D. and Trunfio, P., 2018. Big data systems meet machine learning challenges: towards big data science as a service. *Big data research*, 14, pp.1-11.
- Firliana, R., Kasih, Patmi, 2018. *Algoritma & Pemrograman C++*. Adjie Media Nusantara, Nganjuk.
- Flowers, B.H., 2000. *An introduction to numerical methods in C++*. Oxford University Press.

Gabbrielli, M. and Martini, S., 2023. *Programming languages: principles and paradigms*. Springer Nature.

Goodrich, M.T., Tamassia, R. and Mount, D.M., 2011. *Data structures and algorithms in C++*. John Wiley & Sons.

Goodrich, M. T., Tamassia, R., Mount, D. M., 2011. *Data Structures and Algorithms in C++*. United Kingdom: Wiley.

Halder S, Pal D. 2018. *Data Structures and Algorithms with C*. Alpha Science International Ltd

Hanief, S., Jepriana, I.W. and Kom, S., 2020. *Konsep Algoritme dan Aplikasinya dalam Bahasa Pemrograman C++*. Penerbit Andi.

Hehl, K., 2005. C++ and Java code for recursion formulas in mathematical geodesy. *GPS Solutions*, 9, pp.51-58. DOI: <https://doi.org/10.1007/s10291-004-0116-4>.

Hidayatullah, Priyanto, J. K. K. (2014) *Pemrograman WEB*.

Hindarto dan Eviyanti, A. (2020) *Buku Ajar Algoritma & Pemrograman*. Tersedia pada: <https://b-ok.asia/book/21065648/96b95f>.

I Komang Setia Buana, Gusti Ngurah Mega Nata, I. B. K. S. A. (2018). *STRUKTUR DATA*. Andi Publisher. <https://books.google.co.id/books?id=EH2DDwAAQBAJ&printsec=frontcover#v=onepage&q&f=false>

Indahyanti, U., Rahmawati, Y., 2020. *ALGORITMA DAN PEMROGRAMAN DALAM BAHASA C++*. UMSIDA Press.

Jarvi, J., 1998. *Compile time recursive objects in C++* (pp. 66-77). IEEE. DOI: 10.1109/TOOLS.1998.713588.

Josuttis, N.M., 2012. *The C++ standard library: a tutorial and reference*. Addison-Wesley.

Kadir Sobur Fakultas Ushuluddin IAIN STS Jambi, H. (2015). LOGIKA DAN PENALARAN DALAM PERSPEKTIF ILMU PENGETAHUAN. In *Logika dan Penalaran dalam Perspektif Ilmu Pengetahuan TAJDID: Vol. XIV* (Issue 2).

Kadir, A. (2012). *Algoritma & Pemrograman menggunakan C&C++*. ANDI.

Kadir, A. (2019) *Logika Pemrograman Menggunakan C++*. Pertama. Jakarta: Elex Media Komputindo.

Kanedi, Indra, Yupianti Yupianti, and F. H. U. (2017) *Media Sarana Promosi Makanan Khas Bengkulu Berbasis Website Menggunakan Script PHP*.

Kleiss, R. and van den Oord, G., 2011. CAMORRA: a C++ library for recursive computation of particle scattering amplitudes. *Computer Physics Communications*, 182(2), pp.435-447. DOI: <https://doi.org/10.1016/j.cpc.2010.09.020>.

Kommadi, B., 2019. *Learn Data Structures and Algorithms with Golang: Level Up Your Go Programming Skills to Develop Faster and More Efficient Code*. United Kingdom: Packt Publishing.

Langsam, Y., Augenstein, M.J. and Tenenbaum, A.M., 1996. *Data Structures using C and C++*. Prentice Hall Press.

Liem, Inggriani. 2007. Draft Diktat Kuliah Dasar Pemrograman (Bagian Pemrograman Prosedural). Program Studi Teknik Informatika, Institut Teknologi Bandung. hlm. 141-142

Manjunath.R (2020) *C, C++, Java, Python, PHP, JavaScript and Linux For Beginners*. Manjunath.R.

Martin, E. S., Kennedy, W., Ketelsen, B., 2015. *Go in Action*. United States: Manning.

Mufadhol (2017) *Arsitektur Sistem Komputer (Perangkat Keras)*.

Muhammad Wali, S.T., Efitra, S., Kom, M., Sudipa, I.G.I., Kom, S., Heryani, A., Sos, S., Hendriyani, C., Rakhmadi Rahman, S.T., Kom, M. and Indarto, S.L., 2023. Penerapan & Implementasi Big Data di Berbagai Sektor (Pembangunan Berkelanjutan Era Industri 4.0 dan Society 5.0). PT. Sonpedia Publishing Indonesia.

Munir R. 2011. *Algoritma & Pemrograman Dalam Bahasa Pascal dan C*. Bandung: Informatika Bandung

Mutiara, A. B. (2015) *Panduan Praktis Simulasi Komputer Untuk Sain dan Teknik*.

Oualline, S., 2003. *Practical C++ programming*. " O'Reilly Media, Inc."

Pranata, A. (2005). *Algoritma dan Pemrograman* (3rd ed.). Penerbit Graha Ilmu.

Program, M. (2015). *Program Menghitung Nilai Akhir menggunakan Struct, Fungsi, Array 1, For Dan IF Else*.
<https://majelisprogram.blogspot.com/2015/03/membuat-program-menghitung-nilai-akhir.html>

Rahayuningsih, P. A. (2016) "Analisis Perbandingan Kompleksitas Algoritma Pengurutan Nilai (Sorting)," 4(June).

Rohmah, M.F. and Cahyono, R.E., 2022. *Teori dan Aplikasi Algoritma Pemrograman dengan Bahasa C dan C++*. Penerbit NEM.

Rothermel, G., Harrold, M.J. and Dedhia, J., 2000. Regression test selection for C++ software. *Software Testing, Verification and Reliability*, 10(2), pp.77-109.

Sebesta, R.W., 2012. *Concepts of programming languages*. Pearson Education, Inc.

Sedgewick, R., 2002. *Algorithms in C++*. United Kingdom: Addison-Wesley.

Sedgewick, R., Van Wyk, C. J., 1998. *Algorithms in C++: Fundamentals, data structures, sorting, searching*. Parts 1-4. United Kingdom: Addison-Wesley.

Sedgewick, R., Wayne, K., 2011. *Algorithms*. United Kingdom: Addison-Wesley.

Sianipar, R.H. (2012) *Pemrograman C++ : Dasar Pemrograman Berorientasi Objek*. Andi.

Sianipar, R.H., 2015. *Soal & Penyelesaian C++*. Penerbit Informatika.

Sigit Susanto Putro, Devie Rosa Anamisa, & Fifin Ayu Mufarroha. (2019). *ALGORITMA PEMROGRAMAN* (Tim Media Nusa Creative, Ed.; 1st ed.). Media Nusa Creative.

Singhal, N. and Kumar, A.N., 2000, October. Facilitating problem-solving on nested selection statements using C/C++. In *30th Annual Frontiers in Education Conference. Building on A Century of Progress in Engineering Education. Conference Proceedings (IEEE Cat. No. 00CH37135)* (Vol. 1, pp. T4C-1). IEEE.

Sitorus L, Sembiring DJ. 2012. *Konsep dan Implementasi Struktur Data dengan C++*. Yogyakarta: CV Andi Offset.

Sjukani M. 2012. *Struktur Data (Algoritma dan Struktur Data 2)*, Edisi 5. Jakarta: Mitra Wacana Media

Smith, P., 2004. *Applied data structures with C++*. Jones & Bartlett Learning. Library of Congress-in-Publication Data. Jones and Bartlett Publisher International, Canada.

Stan, E.G., Ciubotaru, D.A., Renda, M. and Alexa, C., 2023. A new advance on dimensional-aware scalar, vector and matrix operations in C++. *Scientific Reports*, 13(1), p.16666.

Standish, T. A., 1995. Data structures, algorithms, and software principles in C. United Kingdom: Addison-Wesley.

Stepanov, A., McJones, P., 2019. Elements of Programming. United States: Semigroup Press.

Stroustrup, B., 1991. The C++ programming language. United Kingdom: Addison-Wesley.

Sugiarto, E. (2019) "Array dan Fungsi," hal. 1–9.

Suhendar, A. (2019) "Struktur Data Sederhana," hal. 2.

Susanto, W.E. and Syukron, A. (2020) *Logika & Algoritma untuk Pemula*. Pertama. Yogyakarta: Graha Ilmu.

Tarkowski, T., 2023. Quilë: C++ genetic algorithms scientific library. *Journal of Open Source Software*, 8(82), p.4902.

Thomas, D.B., 2016, July. Synthesizable recursion for C++ HLS tools. In *2016 IEEE 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP)* (pp. 91-98). IEEE. DOI: 10.1109/ASAP.2016.7760777.

Vivien, V., 2016. Learning Go Programming. United Kingdom: Packt Publishing.

Wadisman, C., & Nozomi, I. (n.d.). *Irohito Nozomi Aplikasi Sistem Informasi Pengolahan Data Penjualan Dan Persediaan Dengan Bahasa Pemrograman Visual Basic.Net 2010*.

Wahyudi, J. (2013). INSTRUKSI BAHASA PEMROGRAMAN ADT (ABSTRACT DATA TYPE) PADA VIRUS DAN LOOP BATCH. *Jurnal Media Infotama*, 9(2).

Weiss, M. A., 2003. Data Structures & Algorithm Analysis in C++. United Kingdom: Pearson Education

Wilis Kaswidjanti, S.Si., M. K. (n.d.). Algoritma dan Pemrograman - Record dan Array of Record. In *SlidePlayer*.

<https://slideplayer.info/slide/12425185/>

Zailani, A. U., Apriyanto, B. dan Zakaria, H. (2020) *Struktur Data*. Tersedia pada: www.unpam.ac.id.

Zakiah, D., Merlina, N., & Mayangky, N. A. (n.d.). *Penerapan Algoritma K-Means Clustering Untuk Mengetahui Kemampuan Karyawan IT*. <http://jurnal.bsi.ac.id/index.php/co-science>

Zein, Afrizal, & E. S. E. (2020) *Algoritma dan Struktur Data, Fakultas Komputer*.

Zhang, H., Luo, J., Hu, M., Yan, J., Zhang, J. and Qiu, Z., 2023, May. Detecting Exception Handling Bugs in C++ Programs. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)* (pp. 1084-1095). IEEE.

BIODATA PENULIS

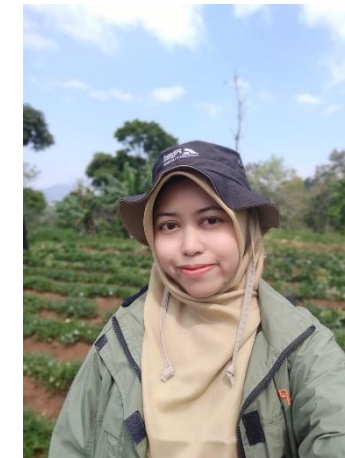


Wika Purbasari, M.Kom adalah seorang Dosen tetap di STMIK Widya Utama. Lahir di kota kecil Purbalingga 22 Juni 1985. Ia lulus pada tahun 2015 hingga mendapatkan gelar Magister Komputer di Universitas Dian Nuswantoro Semarang dibidang ilmu komputer yang merupakan salah satu hobinya. Saat ini ia pun masih aktif sebagai koordinator Kampus mengajar dari awal tahun 2022 dan fokus dalam kegiatan Tri dharma yaitu melakukan tugas dosen mendidik, meneliti dan melakukan pengabdian masyarakat. Penelitian yang didanai oleh Risetdikti pada tahun 2018 berjudul “Model Kepercayaan Konsumen pada Situs E-Commerce di Indonesia: Studi Kasus Kaskus.co.id”. Beberapa penelitian lain yang sudah pernah dilakukan yaitu Pemodelan Augmented Reality Pada Platform Digital Mark Route Berbasis Android (2022) dan Analisis sentimen dan Analisis Jaringan (Network Analysis) Seks Pranikah di Indonesia Menggunakan Data Media Sosial Twitter (2023). Sebagai bentuk pengabdian kepada Masyarakat ia pun terlibat aktif dalam berbagai kegiatan seperti Pelatihan Aplikasi Pengenalan Angka dan Huruf untuk anak TK (2023).



Taufiq Iqbal, menjadi pengajar dan ahli di bidang teknologi informasi, memiliki latar belakang pendidikan dengan gelar Sarjana Komputer (S.Kom.) dari STMIK Ganesha Bandung, lulusan tahun 2014, dan Magister Manajemen (M.M.) dari STIE ISM, lulusan tahun 2015. Berkarya di dunia akademis, Saya telah memberikan kontribusi dalam mengajar berbagai mata kuliah di AMIK dan STMIK Indonesia Banda Aceh, mulai dari Bahasa Pemrograman, Algoritma, Jaringan Komputer, hingga Perancangan Sistem

Informasi. Dengan dedikasi dalam mengembangkan kurikulum dan metodologi pengajaran yang inovatif di bidang informatika, serta komitmen dalam membimbing dan menginspirasi generasi muda di dunia teknologi informasi.



Inayatul Inayah, S.Si., M.Si lahir di Kota Demak pada tanggal 21 Desember 1994. Ia Lulus pada tahun 2020 dan mendapat gelar Magister Fisika dengan konsentrasi Elektronika dan Komputasi di Institut Teknologi Bandung melalui beasiswa LPDP. Sejak tahun 2021, ia telah tercatat sebagai dosen tetap program studi Fisika Institut Teknologi dan Sains Nahdlatul Ulama (ITSNU) Pekalongan. Selain mengajar, ia juga aktif dalam kegiatan tridharma lainnya, yaitu penelitian dan pengabdian. Ia pun pernah diamanahi sebagai ketua Bidang Audit Mutu Internal ITSNU Pekalongan, Reviewer Proposal Penelitian dan Pengabdian ITSNU Pekalongan, Auditor Mutu Internal ITSNU Pekalongan, dan Pembina UKM riset. Saat ini, ia sedang menempuh Pendidikan Doktor di Jurusan Fisika Institut Teknologi Bandung dan menjadi penerima beasiswa BPI-LPDP tahun 2023. Beberapa penelitian dan publikasi yang telah dilakukan berfokus pada bidang elektronika dan instrumentasi, internet of things, dan machine learning. Pada tahun 2022, ia berhasil mendapatkan dana hibah penelitian DIKTI (PDP) dengan judul Rancang Bangun Sistem Pemantauan Panel Surya Secara *Realtime* Berbasis *Internet of Things*, dan pada tahun 2023 ia juga berhasil mendapatkan dana hibah Penelitian Kompetitif Nasional (PT-JH) dengan judul Pengembangan Sistem *Smart Agriculture* untuk Peningkatan Produktivitas Hasil Pertanian Berbasis *Internet of Things* dan *Machine Learning*. Selain aktif melakukan kegiatan penelitian dan pengabdian kepada masyarakat, ia juga aktif sebagai penulis buku dan juga narasumber dalam kegiatan seminar maupun workshop.



Munawir, S.Pd.I., S.Kom., M.M. itulah nama saya. Saya merupakan seorang akademisi yang berpengalaman dan memiliki latar belakang pendidikan gelar Sarjana Pendidikan Agama Islam (S.Pd.I) dari STAI Nusantara Kota Banda Aceh, Sarjana Komputer (S.Kom) dari STMIK Ganesha Bandung, dan gelar Magister Manajemen (M.M.) dari STIE ISM. Saat ini, ia terafiliasi dengan Institut Teknologi dan Bisnis Muhammadiyah Bali pada Program Studi Bisnis Digital. Bidang spesialisasi

Saya meliputi bidang bisnis digital, pendidikan agama, pendidikan moral Islam, aplikasi komputer, e-commerce, sistem manajemen informasi, dan kewirausahaan teknologi. Saya telah mengembangkan karier akademiknya sejak tahun 2006 dan telah mengajar mata kuliah yang beragam, mencakup topik-topik penting dalam disiplin ilmunya. Selain menjadi dosen yang berdedikasi, Saya juga memiliki kontribusi riset yang signifikan. Karyanya terpublikasi dalam jurnal internasional dan mencakup berbagai topik seperti strategi pemasaran berbasis teknologi cerdas, rekomendasi produk e-commerce, manajemen hubungan pelanggan, serta penerapan teknologi dalam pendidikan dan hukum. Dedikasinya terhadap pengembangan ilmu pengetahuan dan teknologi telah mendapatkan pengakuan di komunitas akademik.



Sri Sutjiningtyas, S.Pd., M.Kom., lahir di kota Madiun pada tanggal 27 Januari 1970, menyelesaikan studi jenjang S1 di Institut Keguruan dan Ilmu Pengetahuan (IKIP) Bandung yang sekarang dikenal dengan nama Universitas Pendidikan Indonesia di tahun 1993 pada bidang Pendidikan Matematika. Kemudian tahun 2004 menyelesaikan jenjang S2 bidang Ilmu Komputer di Universitas Gadjah Mada Yogyakarta, hingga mendapat gelar Magister Ilmu Komputer. Penulis menjadi staf dosen sejak

tahun 1994 dan saat ini tercatat sebagai dosen tetap di program studi Teknik Informatika Universitas Nurtanio Bandung. Selain mengajar, penulis aktif membimbing tugas akhir mahasiswa, penelitian dan pengabdian kepada masyarakat. Mata kuliah yang diajarkan adalah mata

kuliah terkait matematika yang relevan dengan bidang ilmu komputer atau informatika, yaitu meliputi matematika dasar, matematika diskrit, aljabar linier, logika matematika dan informatika, metode numerik, statistika dan probabilitas. Adapun karya buku yang telah ditulis pada tahun 2022 adalah Pengantar Teknologi Informatika dan tahun 2023 adalah Statistika dan Probabilitas.



Dr. Erna Hikmawati, S. Kom., M. Kom, OCA, OCP, MOS. Lahir di Kota Bandung pada tanggal 21 Agustus 1992. Ia menempuh pendidikan Strata 1 Teknik Informatika di Universitas Nasional PASIM dan lulus pada tahun 2013. Gelar Magister Komputer didapatkan pada tahun 2017 dari program studi Sistem Informasi di STMIK LIKMI. Pada tahun 2023, ia berhasil menyelesaikan pendidikan Doktor di Sekolah Teknik Elektro dan Informatika ITB dengan

beasiswa LPDP. Saat ini ia tercatat sebagai dosen tetap di Program Studi D3 Rekayasa Perangkat Lunak Aplikasi, Fakultas Ilmu Terapan, Telkom University.



Fauzan Natsir, M.Kom lahir di Kota Surakarta pada tahun 1991. Fauzan lulus S1 pada tahun 2013 di Universitas Muhammadiyah Surakarta (UMS) dan lulus S2 pada tahun 2019 di Universitas Islam Indonesia (UII). Saat ini, tercatat sebagai dosen tetap di Program Studi Teknik Informatika Universitas Indraprasta PGRI. Selain mengajar, beliau juga menjadi tenaga pengajar dengan skema Junior Web Developer dan Aplikasi Perkantoran di

BPPTIK Kementerian Kominfo. Serta sebagai Asesor Kompetensi di bidang Junior Web Developer yang terakui oleh BNSP. Penulis fokus terhadap bidang Digital Forensic, Software Engineering, Decision Support System, dan Knowledge Management System. Penulis dapat dihubungi melalui LinkedIn "Fauzan Natsir"; instagram @fauzannatsir atau email di files.fauzan@gmail.com.



Anak Agung Sandatya Widhiyanti, S.Kom., M.Kom. lahir di Kota Denpasar Provinsi Bali pada tanggal 28 Maret 1997. Penulis menempuh pendidikan Sarjana Sistem Informasi di STMIK STIKOM Bali yang sekarang sudah berubah bentuk menjadi Institut Teknologi dan Bisnis STIKOM Bali. Penulis lulus pada tahun 2018 kemudian melanjutkan pendidikan Magister Ilmu Komputer di Universitas Pendidikan Ganesha dan lulus pada tahun 2021. Saat ini

penulis berprofesi sebagai dosen tetap dan mengampu mata kuliah pemrograman, rekayasa perangkat lunak, praktikum basis data, analisa perancangan sistem informasi, sistem pakar dan teknik peramalan di Institut Teknologi dan Bisnis STIKOM Bali. Selain menjadi dosen, penulis juga berprofesi sebagai guru produktif untuk program keahlian pengembangan perangkat lunak dan gim di salah satu sekolah menengah kejuruan yang berlokasi di kabupaten Badung, Provinsi Bali. Penulis menguasai beberapa bahasa dan teknik pemrograman di antaranya PHP, MySQL, CSS, CodeIgniter, C#, C++, dan Java. Selain itu, penulis juga mampu mendokumentasikan program ke dalam *diagram* seperti UML, DFD, ERD, BPMN, Basis Data Konseptual dan *Flowchart*. Penulis dapat dihubungi melalui email: widhiyanti@stikom-bali.ac.id



Muhammad Wali, Menjadi dosen sejak awal tahun 2016, saya mulai mengembangkan minat dan keahlian dalam penulisan di bidang ilmu komputer. Sebelumnya, saya menyelesaikan pendidikan di SMKN 1 Kota Sabang dan melanjutkan studi Diploma III di AMIK Indonesia, jurusan Manajemen Informatika, lulus pada tahun 2008. Kemudian, saya melanjutkan pendidikan sarjana di Universitas Serambi Mekkah, mengambil program studi Teknik Informatika, dan

berhasil menyelesaikan studi pada tahun 2011. Seterusnya, saya melanjutkan studi S2 di bidang Manajemen Sistem Informasi di STIE-ISM Tangerang dan meraih gelar pada tahun 2015. Kepakaran saya mencakup berbagai bidang ilmu komputer, termasuk Data Science, Software

Development, Expert Systems, Mobile Development, dan Education Technology. Sebagai seorang profesional di dunia pendidikan, saya aktif sebagai peneliti di bidang keahlian tersebut. Beberapa penelitian yang telah saya lakukan mendapatkan dukungan dana baik dari internal perguruan tinggi maupun Kemenristek DIKTI. Saya juga berpartisipasi dalam kegiatan pengabdian masyarakat, dengan fokus memberikan kontribusi pada pemanfaatan teknologi untuk meningkatkan ekonomi lokal dan nasional, berkolaborasi dengan berbagai perguruan tinggi di Indonesia. Dengan upaya ini, saya bertujuan mewujudkan karir sebagai seorang dosen yang profesional dan berdedikasi.



M. Syauqi Haris, M.Kom. lahir di Kota Lamongan Jawa Timur pada 22 Februari 1985. Lulus dari Sarjana Ilmu Komputer Universitas Brawijaya pada tahun 2010 dan lulus dari Magister Ilmu Komputer Universitas Brawijaya pada tahun 2020. Berpengalaman lebih 10 tahun dalam industri pengembangan perangkat lunak dan saat ini aktif sebagai dosen prodi S1 Informatika di Institut Teknologi, Sains, dan Kesehatan (ITSK) RS dr. Soepraoen Malang.

Selain menjadi dosen, penulis aktif sebagai CEO PT Narasumber Teknologi Indonesia, sebuah startup di bidang konsultan teknologi informasi dan digital marketing bagi koperasi dan UMKM sehingga seringkali diundang sebagai narasumber oleh berbagai instansi negeri maupun swasta sebagai trainer di berbagai acara bimbingan teknis maupun pelatihan bagi koperasi dan UMKM. Dalam penelitian, penulis aktif melakukan riset di bidang rekayasa perangkat lunak dan pembelajaran mesin. Pada tahun 2022, menerima hibah Penelitian Dosen Pemula (PDP) dengan judul Prediksi Prevalensi Stunting di Provinsi Jawa Timur dengan menggunakan Algoritma *Machine Learning*.



Hasan Basri, S.Kom, M.Kom, adalah seorang Dosen dengan latar belakang pendidikan yang kuat di bidang Ilmu Komputer. Ia meraih gelar sarjana (S.Kom) dari Universitas Mercu Buana pada tahun 2016, dan kemudian melanjutkan pendidikan tingkat magister (M.Kom) di Institut Pertanian Bogor pada tahun 2018. Selama lebih dari 4,5 tahun, Hasan Basri

telah menjadi seorang dosen yang berdedikasi, mengajar berbagai mata kuliah terkait Sistem Informasi, termasuk algoritma pemrograman, pemrograman web, dan pemrograman berorientasi objek (OOP). Selain sebagai pendidik yang berpengalaman, Hasan Basri juga memiliki minat dan pengalaman dalam berbagai proyek dan penelitian di bidang Sistem Informasi. Kombinasi antara pendidikan yang solid dan pemahaman yang mendalam tentang teknologi informasi memungkinkan Hasan Basri untuk berperan aktif dalam pengembangan teknologi dan pendidikan. Dengan semangat untuk terus belajar dan berinovasi, ia berusaha untuk berkontribusi pada kemajuan bidang Sistem Informasi dan berbagi pengetahuannya dengan generasi mendatang.

ALGORITMA PEMROGRAMAN

Dalam dunia yang terus berubah, di mana teknologi informasi berkembang dengan pesat, pemahaman yang mendalam tentang algoritma pemrograman menjadi kunci untuk siapapun yang memasuki dunia informatika dan ilmu komputer. Buku "Algoritma Pemrograman" ini dirancang khusus sebagai panduan esensial bagi siapa saja yang ingin menguasai dasar-dasar informatika dan ilmu komputer. Buku ini unik karena tidak hanya menyediakan teori yang mendalam tentang algoritma, tapi juga menggabungkannya dengan sketsa algoritma dalam bentuk *flowchart* dan *pseudocode*, serta implementasinya dalam bahasa pemrograman C++. Kami mengakui bahwa banyak buku telah membahas algoritma pemrograman, namun buku ini dibuat untuk tetap relevan dengan perkembangan teknologi informasi terkini, sehingga pembaca dapat selalu berada di garis depan pengetahuan.

Setiap bab dalam buku ini dirancang untuk memperdalam pemahaman pembaca, mulai dari Algoritma dan Flowchart, Tipe Data, Perulangan, Penyeleksian, Prosedur, Fungsi, Array, Record, hingga konsep lanjutan seperti Rekursi, serta Algoritma Pencarian dan Pengurutan. Dengan penjelasan teori yang detail, diikuti oleh sketsa algoritma dalam bentuk *flowchart* dan *pseudocode*, serta contoh implementasi kode dalam C++, buku ini menawarkan pandangan holistik tentang bagaimana algoritma bekerja dan diaplikasikan dalam dunia nyata. Kami berharap buku ini tidak hanya menjadi sumber pengetahuan yang berharga bagi pelajar dan mahasiswa, tetapi juga menjadi inspirasi bagi siapa saja yang ingin mengeksplorasi dan berinovasi dalam dunia pemrograman dan teknologi informasi.



Penerbit Haura Utama

• Jarakta KRI, Jawa Barat
• Instagram: @haura_utama
• Website: penerbit.haura.com
• Email: haura@haura.com

ISBN 978-623-492-783-2

