



Logika dan Algoritma Pemrograman

Fondasi Dasar Menjadi Programmer Andal



**M Rhifky Wayahdi, Phie Chyan, Agung Yuliyanto Nugroho,
Amril Mutoi Siregar, Hansi Effendi, Satriawaty Mallu, Samsuriah,
Dianta Ginting, Ni Putu Linda Santiari, Ismi Rizqa Lina, Waris Marsisno,
Fahmi Ruziq, Mardiah, Anis Fitri Nur Masruriyah, Santi Prayudani,
Sri Retnowati, Ari Widiastono, Dita Novita Sari,
A.Kachsyfur Djasim Ilyas Paenrongi, Triana Dewi Salma**

Logika dan Algoritma Pemrograman

Fondasi Dasar Menjadi Programmer Andal

**M Rhifky Wayahdi, Phie Chyan, Agung Yuliyanto Nugroho,
Amril Mutoi Siregar, Hansi Effendi, Satriawaty Mallu,
Samsuriah, Dianta Ginting, Ni Putu Linda Santiari, Ismi
Rizqa Lina, Waris Marsisno, Fahmi Ruziq, Mardiah, Anis
Fitri Nur Masruriyah, Santi Prayudani, Sri Retnowati, Ari
Widiastono, Dita Novita Sari, A.Kachsufur Djasim Ilyas
Paenrongi, Triana Dewi Salma**



PT. MIFANDI MANDIRI DIGITAL

Undang-Undang No. 28 Tahun 2014 Tentang Hak Cipta:

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 Rp. 100.000.000,- (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 Rp. 500.000.000,- (lima ratus juta rupiah).
3. 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 a, huruf b, huruf e, dan/atau huruf g untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp. 1.000.000.000,- (satu miliar rupiah).
4. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp. 4.000.000.000,- (empat miliar rupiah).

Logika dan Algoritma Pemrograman Fondasi Dasar Menjadi Programmer Andal

M Rhifky Wayahdi, Phie Chyan, Agung Yuliyanto Nugroho, Amril Mutoi Siregar, Hansi Effendi, Satriawaty Mallu, Samsuriah, Dianta Ginting, Ni Putu Linda Santiari, Ismi Rizqa Lina, Waris Marsisno, Fahmi Ruziq, Mardiah, Anis Fitri Nur Masruriyah, Santi Prayudani, Sri Retnowati, Ari Widiastono, Dita Novita Sari, A.Kachsufur Djasim Ilyas Paenrongi, Triana Dewi Salma

ISBN: 978-634-7226-25-9

Editor : Sarwandi, M.Pd.T
Layout : Miftahul Jannah, M.Kom
Desain sampul : Rifki Ramadan

Penerbit
PT. Mifandi Mandiri Digital

Redaksi & Distributor Tunggal
PT. Mifandi Mandiri Digital
Komplek Senda Residence Jl. Payanibung Ujung D Dalu
Sepuluh-B Tanjung Morawa Kab. Deli Serdang Sumatera Utara

Cetakan Pertama, Juli 2025

Hak Cipta © 2025 by PT. Mifandi Mandiri Digital

Hak cipta Dilindungi Undang-Undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin tertulis dari penerbit.

Kata Pengantar

Di era transformasi digital saat ini, keterampilan dalam logika dan algoritma pemrograman menjadi kemampuan dasar yang sangat penting, bukan hanya bagi calon programmer, tetapi juga bagi siapa pun yang ingin memahami cara kerja teknologi di balik aplikasi, sistem, dan perangkat digital yang digunakan sehari-hari. Logika dan algoritma bukan sekadar teori atau konsep matematis; keduanya adalah fondasi dalam proses berpikir sistematis, pemecahan masalah, dan pengembangan perangkat lunak yang efisien dan andal.

Buku ini hadir sebagai panduan komprehensif yang dirancang untuk membantu pembaca memahami prinsip-prinsip dasar logika dan algoritma pemrograman secara bertahap dan terstruktur. Dimulai dari pengenalan logika dasar, struktur algoritma, hingga penerapan konsep lanjutan seperti rekursi, struktur data, dan algoritma pencarian dan pengurutan, buku ini dilengkapi dengan studi kasus nyata dan ilustrasi yang memudahkan pemahaman. Tidak hanya berfokus pada teori, buku ini juga membimbing pembaca untuk menerapkan konsep algoritmik ke dalam bahasa pemrograman Python secara praktis.

Dengan cakupan yang luas namun disajikan dengan bahasa yang mudah dipahami, buku ini diharapkan menjadi referensi utama bagi pelajar, mahasiswa, guru, maupun siapa saja yang ingin membangun fondasi kuat dalam dunia pemrograman. Semoga kehadiran buku ini dapat membuka jalan bagi pembaca untuk berpikir logis, menyusun solusi secara

algoritmik, dan mengembangkan kemampuan teknis yang dibutuhkan untuk menjadi programmer andal di era digital yang penuh tantangan dan peluang.

Medan, Maret 2025

Penulis

Daftar Isi

| | |
|---|-----------|
| Kata Pengantar | i |
| Daftar Isi | iii |
| BAB 1 PERAN LOGIKA DAN ALGORITMA DALAM PEMROGRAMAN .. | 1 |
| Pendahuluan | 1 |
| Pengertian Logika dalam Pemrograman | 2 |
| Pengertian Algoritma dalam Pemrograman | 4 |
| Hubungan antara Logika dan Algoritma | 5 |
| Peran Logika dan Algoritma dalam Pemecahan Masalah | 6 |
| Contoh Penerapan Logika dan Algoritma dalam Pemrograman | 8 |
| BAB 2 KONSEP DASAR LOGIKA: PROPOSISI, PREDIKAT DAN | |
| PENALARAN | 11 |
| Pendahuluan | 11 |
| Pengertian dan Sejarah Singkat Logika | 12 |
| Proposisi: Definisi dan Operasi Dasar | 13 |
| Logika Predikat dan Kuantifikasi | 15 |
| Tantangan Implementasi | 18 |
| Validitas dan Kesesatan Logika | 19 |
| BAB 3 JENIS-JENIS LOGIKA DALAM PEMOGRAMAN | 21 |
| Pendahuluan | 21 |
| Logika Kondisional (Conditional Logic) | 21 |
| Logika Perulangan (Looping Logic) | 22 |
| Logika Boolean (Boolean Logic) | 23 |
| Logika Rekursif (Recursive Logic) | 25 |
| Logika Kombinatorial (Combinatorial Logic) | 25 |
| BAB 4 STRUKTUR DASAR ALGORITMA: DEFENISI DAN | |
| KARAKTERISTIK | 28 |
| Pendahuluan | 28 |
| Definisi Algoritma | 31 |
| Karakteristik Algoritma | 35 |

| | |
|---|-----|
| BAB 5 NOTASI ALGORITMIK: PSEUDOCODE DAN FLOWCHART | 39 |
| Pendahuluan | 39 |
| Pengertian Pseudocode | 40 |
| Struktur Dasar Pseudocode | 41 |
| Contoh Pseudocode | 42 |
| Pengertian Flowchart | 43 |
| Simbol-Simbol dalam Flowchart | 45 |
| Contoh Flowchart | 46 |
| Perbandingan Flowchart dan Pseudocode | 47 |
| | |
| BAB 6 STRUKTUR KONTROL DALAM ALGORITMA: SEQUENCING, SELECTION, ITERATION | 50 |
| Pendahuluan | 50 |
| Struktur Kontrol Algoritma Sequencing | 51 |
| Struktur Kontrol Algoritma Selection | 57 |
| Struktur Kontrol Algoritma Iteration | 60 |
| | |
| BAB 7 TIPE DATA DAN VARIABEL DALAM ALGORITMA | 67 |
| Pendahuluan | 67 |
| Mengetahui tentang Tipe Data | 68 |
| Variabel | 69 |
| | |
| BAB 8 OPERASI ARITMETIKA DAN LOGIKA DALAM ALGORITMA | 74 |
| Pendahuluan | 74 |
| Operasi Aritmetika | 74 |
| Logika dalam Pemrograman | 78 |
| | |
| BAB 9 PERCABANGAN (DECISION MAKING) DALAM ALGORITMA ... | 85 |
| Pendahuluan | 85 |
| Pengertian dan Fungsi Percabangan | 86 |
| Jenis-jenis Struktur Percabangan | 87 |
| Notasi Algoritma dan Flowchart untuk Percabangan | 89 |
| Implementasi Percabangan dalam Bahasa Pemrograman | 92 |
| Studi Kasus & Latihan | 94 |
| Kesalahan Umum dalam Percabangan | 96 |
| | |
| BAB 10 PERULANGAN (LOOPING) DAN PENERAPANNYA | 101 |
| Pendahuluan | 101 |
| Pengenalan Looping | 102 |
| For Loops | 104 |
| For Bersarang | 106 |

| | |
|--|-----|
| BAB 11 PENGENALAN STRUKTUR DATA DALAM ALGORITMA | 116 |
| Pendahuluan | 116 |
| Prinsip Dasar Struktur Data | 117 |
| Jenis (Klasifikasi) Struktur Data | 119 |
| BAB 12 ARRAY DAN PENGGUNAANNYA DALAM PEMROGRAMAN | 126 |
| Pendahuluan | 126 |
| Pengertian dan Karakteristik Array | 127 |
| Deklarasi dan Inisialisasi Array | 128 |
| Operasi Dasar pada Array | 129 |
| Array Satu Dimensi dan Multidimensi | 131 |
| Keunggulan dan Keterbatasan Array | 135 |
| Contoh Implementasi Array | 138 |
| BAB 13 STRING DAN OPERASINYA DALAM ALGORITMA | 140 |
| Pendahuluan | 140 |
| Operasi Dasar String | 140 |
| Struktur Data untuk Menyimpan String | 142 |
| BAB 14 STRUKTUR DATA LIST, STACK DAN QUEUE | 146 |
| Pendahuluan | 146 |
| List (Daftar) | 147 |
| Stack (Tumpukan) | 152 |
| Queue (Antrian) | 154 |
| BAB 15 ALGORITMA SORTING: BUBBLE SORT, SELECTION SORT, INSERTION SORT | 157 |
| Pendahuluan | 157 |
| Bubble Sort | 158 |
| Selection Sort | 161 |
| Insertion Sort | 164 |
| BAB 16 PENGENALAN REKURSI DALAM ALGORITMA | 168 |
| Pendahuluan | 168 |
| Pengertian Rekursi | 169 |
| Komponen Rekursi | 170 |
| Perbandingan Rekursi dan Iterasi | 171 |
| Contoh Rekursi | 172 |
| Kelebihan dan Kekurangan Rekursi | 174 |
| Studi Kasus & Latihan Soal | 175 |

| | |
|--|-----|
| BAB 17 ALGORITMA PENCARIAN: LINEAR SEARCH DAN BINARY SEARCH | 180 |
| Pendahuluan | 180 |
| Linear Search | 181 |
| Binary Search | 184 |
| Perbandingan Linear dan Binary Search | 188 |
| | |
| BAB 18 PEMROGRAMAN MODULAR: FUNGSI DAN PROSEDUR | |
| DALAM ALGORITMA | 192 |
| Pendahuluan | 192 |
| Dasar-Dasar Pemrograman Modular | 193 |
| Fungsi: Definisi, Struktur, dan Penerapan | 194 |
| Prosedur: Definisi, Struktur, dan Penerapan | 196 |
| Perbandingan Fungsi dan Prosedur | 198 |
| Konsep Parameter dan Scope | 199 |
| Teknik Pengembangan Modul yang Baik | 200 |
| Studi Kasus dan Penerapan dalam Proyek Nyata | 201 |
| Konsep Lanjutan: Rekursi dan Modularitas | 203 |
| Best Practices dan Tantangan dalam Pemrograman Modular | 205 |
| Rangkuman dan Arahkan Pengembangan Lebih Lanjut | 205 |
| | |
| BAB 19 PEMROSESAN DATA DENGAN ALGORITMA GREEDY & DIVIDE AND CONQUER | 207 |
| Pendahuluan | 207 |
| Algoritma Greedy (Greed Algorithm) | 211 |
| Kelebihan dan Kekurangan Algoritma Greedy | 213 |
| Algoritma Divide and Conquer | 215 |
| Kelebihan dan Kekurangan Algoritma Divide-and-Conquer | 217 |
| Studi Kasus Algoritma Greedy: Kompresi Data Huffman | 220 |
| Studi Kasus Algoritma Divide-and-Conquer: Merge Sort untuk Pengurutan Data Besar | 228 |
| Efektivitas dan Efisiensi: Perbandingan Greedy vs Divide-and-Conquer .. | 234 |
| | |
| BAB 20 PENGENALAN BAHASA PEMROGRAMAN DAN IMPLEMENTASI ALGORITMA | 240 |
| Pendahuluan | 240 |
| Mengenal Bahasa Pemrograman Python | 241 |
| Python vs Bahasa Lain | 242 |
| Menerjemahkan Algoritma ke Python | 244 |
| Implementasi Algoritma Populer dalam Python | 246 |
| Praktik Baik dalam Pengkodean Algoritma dengan Python | 249 |

Daftar Pustaka 252
Tentang Penulis 262

BAB 1 PERAN LOGIKA DAN ALGORITMA DALAM PEMROGRAMAN

Pendahuluan

Dalam dunia pemrograman, logika dan algoritma merupakan dua konsep dasar yang tidak dapat dipisahkan. Keduanya menjadi pondasi utama dalam pengembangan perangkat lunak dan aplikasi. Tanpa pemahaman yang baik tentang logika dan algoritma, seorang *programmer* akan kesulitan dalam menyelesaikan masalah secara sistematis dan efisien. Logika memberikan kerangka berpikir sistematis untuk merancang solusi, sementara algoritma menyediakan langkah-langkah terstruktur untuk mencapai tujuan tersebut (Mirkowska & Salwicki, 1987). Bab ini akan membahas peran penting logika dan algoritma dalam pemrograman, serta bagaimana keduanya saling melengkapi untuk menciptakan solusi yang efektif.

Logika berfungsi sebagai alat untuk menilai argumen dan membedakan antara penalaran yang benar dan salah. Dalam konteks ilmu komputer, logika tidak hanya membantu dalam memahami dan merancang algoritma, tetapi juga menjadi dasar bagi pengembangan program komputer yang efisien. C. A. R. Hoare, seorang tokoh terkemuka dalam bidang ini, menekankan bahwa program komputer dapat dipandang sebagai ekspresi matematis, di mana semua sifat program dapat dieksplorasi melalui penalaran deduktif. Dengan demikian,

BAB 3 JENIS-JENIS LOGIKA DALAM PEMOGRAMAN

Pendahuluan

Dalam pemrograman, logika berperan penting dalam pengambilan keputusan dan alur eksekusi suatu program. Berikut adalah beberapa jenis logika yang umum digunakan dalam pemrograman.

Logika Kondisional (*Conditional Logic*)

Logika kondisional sering kali diterapkan menggunakan pernyataan seperti *if-else*, *Switch-Case*, atau ekspresi ternary dalam berbagai bahasa pemrograman. Dengan adanya struktur ini, program dapat menangani berbagai skenario tanpa harus mengeksekusi semua bagian kode secara berurutan.

Contoh Kasus: Menentukan Kategori Usia

Sebuah program diminta untuk mengelompokkan seseorang ke dalam kategori usia berdasarkan umur yang dimasukkan oleh pengguna. Kategorinya sebagai berikut.

Anak-anak: 0 - 12 tahun

Remaja: 13 - 17 tahun

Dewasa: 18 - 59 tahun

Lansia: 60 tahun ke atas

Berikut adalah implementasi dalam bahasa Python menggunakan *if-elif-else*:

BAB 4 STRUKTUR DASAR ALGORITMA: DEFENISI DAN KARAKTERISTIK

Pendahuluan

Algoritma dapat dicirikan sebagai kumpulan prosedur atau arahan yang metodis dan berurutan yang bertujuan menyelesaikan masalah tertentu atau melaksanakan tugas yang ditunjuk. Dalam bidang ilmu komputer, algoritma merupakan pilar fundamental rekayasa perangkat lunak, manipulasi data, dan optimalisasi beragam sistem. Mereka menemukan aplikasi di banyak disiplin ilmu, mencakup domain seperti pengambilan informasi, enkripsi data, dan kecerdasan buatan. Seperti yang diartikulasikan oleh Donald Knuth, seorang tokoh terkemuka di bidang ilmu komputer, algoritma harus menunjukkan atribut penting tertentu, termasuk keterbatasan, kepastian, input, *output*, dan efektivitas. Dengan memenuhi kriteria ini, algoritma dapat digunakan secara efektif untuk mengatasi berbagai tantangan dalam domain komputasi (Halimatussyah Purba & Yahfizham Yahfizham, 2023).

Agar suatu algoritma dapat digunakan dengan kemanjuran dalam penyelesaian masalah, ia harus memiliki beberapa karakteristik penting. Pertama, algoritma harus diartikulasikan dengan jelas dan didefinisikan secara komprehensif. Ini mengharuskan langkah-langkah prosedural dapat ditafsirkan oleh agen manusia dan sistem komputasi tanpa tingkat ambiguitas apa pun. Kedua, algoritma harus terorganisir dan logis, memastikan bahwa itu menghasilkan

BAB 5 NOTASI ALGORITMIK: *PSEUDOCODE* DAN *FLOWCHART*

Pendahuluan

Notasi algoritmik adalah cara untuk merepresentasikan langkah-langkah penyelesaian masalah (algoritma) sebelum diimplementasikannya ke dalam kode program (Cormen et al., 2022). Dalam pemrograman, notasi ini membantu *programmer* merancang solusi secara sistematis, memahami alur logika, dan berkomunikasi dengan tim (Hazzan & Lapidot, 2021). Ada dua jenis notasi algoritmik yang umum digunakan:

1. *Pseudocode*: Representasi berbasis teks yang menyerupai bahasa pemrograman tetapi lebih sederhana dan tidak terikat pada aturan sintaks bahasa tertentu (Chen & Wang, 2021).
2. *Flowchart*: Representasi grafis yang menggunakan simbol-simbol untuk menggambarkan alur proses (Alotaibi & Alotaibi, 2020).

Keduanya memiliki kelebihan masing-masing: *flowchart* lebih visual dan mudah dipahami secara intuitif, sedangkan *Pseudocode* lebih fleksibel dan cocok untuk dokumentasi teknis (Luxton-Reilly & Denny, 2020). Dalam bab ini, kita akan mempelajari kedua jenis notasi ini, bagaimana cara membuatnya, serta kapan menggunakannya dalam pengembangan algoritma.

Berikut ini adalah contoh menentukan bilangan genap atau ganjil menggunakan *Pseudocode* dan *flowchart*.

BAB 6 STRUKTUR KONTROL DALAM ALGORITMA: SEQUENCING, SELECTION, ITERATION

Pendahuluan

Struktur kontrol dalam algoritma digunakan oleh para pengembang perangkat lunak, *programmer*, dan siapa saja yang merancang atau menulis program komputer. Mereka menggunakan struktur kontrol untuk mengatur alur eksekusi program agar dapat menjalankan tugas secara efisien dan sesuai dengan kebutuhan.

Misalnya, seorang *programmer* yang membuat aplikasi atau sistem informasi akan menggunakan struktur kontrol seperti percabangan (*if-else*) dan pengulangan (*loop*) untuk mengatur proses pengolahan data, pengambilan keputusan otomatis, atau pengulangan tugas tertentu. Selain *programmer*, pelajar yang belajar pemrograman juga mempelajari struktur kontrol ini sebagai bagian dari dasar-dasar pemrograman untuk membangun algoritma yang logis dan efisien.

Secara umum, siapa pun yang mengembangkan algoritma atau menulis kode program—baik itu *programmer* profesional, mahasiswa, pengembang aplikasi, maupun pengembang sistem—menggunakan struktur kontrol dalam algoritma mereka. Fungsi utama penggunaannya adalah untuk memastikan program dapat berjalan secara dinamis dan mampu menangani berbagai situasi serta kondisi yang berbeda.

Struktur kontrol merupakan unsur penting dalam

BAB 7 TIPE DATA DAN VARIABEL DALAM ALGORITMA

Pendahuluan

Bab ini mengenalkan berbagai fondasi Bahasa pemrograman baik Bahasa pemrograman c/c++ maupun Bahasa pemrograman java seperti tipe data, variabel. Di dalam lingkungan para pemrograman. Kita akan menganalisis tipe data, variabel dalam lingkup algoritma pemrograman.

Pengenal (Identifier) adalah suatu nama yang digunakan dalam program untuk menyatakan variabel, fungsi, dan lain-lain. Aturan umum yang berlaku bagi pengenal adalah sebagai berikut.

1. Diawali dengan huruf atau simbol garis-bawah ().
2. Penggunaan yang lain seperti huruf, angka, ataupun simbol underline / garis-bawah ().
3. Ada perbedaan antara huruf kecil dan besar (Kadir, 2012).

Pada implementasi dan latihan, ada kaidah tidak tertulis dalam pemberian nama. Misalnya, nama variabel dengan menggunakan permulaan huruf kecil dan pada awal kata berikutnya ditulis menggunakan huruf besar. Penggunaan nama konstanta dituliskan sepenuhnya dengan huruf besar.

Pada ciri-ciri yang sempurna dan yang *error* dapat ditampilkan pada tabel berikut ini.

BAB 9 PERCABANGAN (*DECISION MAKING*) DALAM ALGORITMA

Pendahuluan

Dalam dunia pemrograman dan algoritma, kemampuan untuk membuat keputusan berdasarkan kondisi tertentu merupakan aspek fundamental. Sebuah program tidak hanya berjalan secara linier dari awal hingga akhir, tetapi sering kali memerlukan percabangan untuk menentukan tindakan yang tepat sesuai dengan situasi yang terjadi. Di sinilah peran penting struktur percabangan dalam algoritma: memungkinkan program untuk "berpikir" dan menyesuaikan perilakunya berdasarkan masukan atau keadaan tertentu.

Struktur percabangan dikenal juga dengan istilah *decision making*, yaitu proses memilih satu dari beberapa jalur eksekusi berdasarkan hasil evaluasi kondisi logis. Misalnya, dalam kehidupan sehari-hari kita sering mengambil keputusan: jika hari hujan, maka kita membawa payung; jika tidak, kita pergi seperti biasa. Logika yang sama diterapkan dalam algoritma, di mana komputer akan menjalankan perintah yang berbeda tergantung dari hasil kondisi yang diberikan.

Melalui bab ini, pembaca akan mempelajari berbagai bentuk struktur percabangan seperti *if*, *if-else*, *if-else if-else*, dan *Switch-Case*. Selain itu, akan dibahas pula bagaimana menuliskan percabangan dalam notasi algoritma, *flowchart*, hingga implementasinya dalam bahasa pemrograman. Dengan memahami materi ini, pembaca diharapkan mampu membuat

BAB 10 PERULANGAN (*LOOPING*) DAN PENERAPANNYA

Pendahuluan

Ketika masih muda pasti pernah diajari tabel angka. Tabel nomor memiliki pola. Menulis tabel dalam suatu ujian memerlukan penulisan, katakanlah, $n \times$ diikuti dengan i (i bervariasi dari 1 sampai n) dan kemudian hasil perhitungannya (yaitu $n \times 1$, $n \times 2$, dan seterusnya). Banyak situasi seperti itu mengharuskan untuk mengulang tugas yang diberikan berkali-kali.

Pengulangan ini dapat digunakan untuk menghitung nilai suatu fungsi, untuk mengeprint suatu pola atau untuk sekadar mengulang sesuatu. Bab ini membahas *loop* dan iterasi yang merupakan bagian integral dari pemrograman prosedural. *Looping* berarti mengulang satu set pernyataan sampai suatu kondisi benar. Berapa kali set ini diulang tergantung pada kondisi pengujian. Juga, apa yang harus diulang perlu dicatat dengan pertimbangan matang.

Secara umum, pengulangan sebuah blok memerlukan hal-hal berikut.

1. Memutuskan apa yang akan diulang: himpunan pernyataan.
2. Kondisi pengujian atau berapa kali himpunan pernyataan harus diulang.
3. Kasus khusus di mana *loop* putus (atau berlanjut, keluar dari pernyataan tertentu).

BAB 11 PENGENALAN STRUKTUR DATA DALAM ALGORITMA

Pendahuluan

Secara umum komputer mengenal berbagai jenis data, misalnya integer (bilangan bulat), karakter, pecahan dan sebagainya. Namun demikian untuk merepresentasikan suatu entitas di dunia nyata sering kali diperlukan penggunaan berbagai jenis data sekaligus. Misal untuk mengetahui karakteristik setiap pelanggan dari suatu gerai penjualan minuman sekurang-kurangnya diperlukan informasi tentang jenis kelamin (bisa berupa karakter atau integer) dan usia (integer) dari setiap pelanggan. Untuk keperluan yang lebih kompleks akan lebih banyak lagi jenis data yang diperlukan. Dalam konteks pemrograman atau pengolahan data dengan komputer, data dari suatu entitas seperti yang telah diuraikan dapat terdiri dari dua atau lebih elemen data. Setiap elemen data merepresentasikan suatu karakteristik tertentu.

Komputer pada dasarnya hanyalah sebuah sistem atau mesin. Komputer tidak akan secara otomatis mampu mengenali suatu elemen merupakan anggota dari data yang mana. Untuk itulah setiap data dan elemen-elemennya yang disimpan atau digunakan komputer perlu diatur. Untuk pengaturan inilah diperlukan struktur data. Secara umum, struktur data dapat diartikan sebagai cara khusus untuk menyimpan dan mengatur data dalam komputer sehingga dapat digunakan secara efisien. Sebenarnya struktur data merupakan susunan logis dari elemen-

BAB 12 ARRAY DAN PENGGUNAANNYA DALAM PEMROGRAMAN

Pendahuluan

Dalam pemrograman, struktur data merupakan komponen penting yang memungkinkan pengelolaan data secara efisien. *Array* adalah struktur data yang paling umum dan mudah digunakan (Izadkhah, 2022). Dengan *array*, *programmer* dapat menyimpan sekumpulan data dengan tipe yang sama dalam urutan tertentu, yang membuat akses dan perubahan data lebih mudah. Dengan kemampuannya yang fleksibel, *array* menjadi fondasi bagi banyak algoritma dan aplikasi pemrograman.

Struktur data *array* menyimpan elemen dalam urutan yang diindeks dan memungkinkan setiap elemen untuk diakses secara langsung melalui indeksnya (Xtracts, 2023). Keunggulan utama *array* adalah kemampuannya untuk mengakses elemen secara acak dalam waktu konstan, yaitu $O(1)$, yang berarti waktu akses tidak tergantung pada ukuran *array*. Hal ini membuat *array* sangat efisien untuk operasi seperti pengambilan data atau modifikasi elemen.

Array tidak hanya terbatas pada penyimpanan data satu dimensi. Dalam banyak kasus, *array* multidimensi digunakan untuk merepresentasikan data yang lebih kompleks, seperti matriks atau tabel (Sharma, 2025). Misalnya, *array* dua dimensi sering digunakan dalam pemrosesan gambar, di mana setiap elemen mewakili piksel. Selain itu, *array* juga menjadi

BAB 13 *STRING* DAN OPERASINYA DALAM ALGORITMA

Pendahuluan

Pada bab ini, akan dibahas mengenai definisi *string*, operasi dasar pada *string*, struktur data pendukung *String*, dan perbandingan antar struktur data yang dapat menyimpan *string*. Kita akan mulai dari definisi *string*.

String adalah suatu tipe data non numerik yang tersusun dari satu atau kumpulan char atau *substring*. *String* dapat berupa huruf (a-z, A-Z), angka (0-9), tanda baca (@, #, dan lain-lain), spasi () (Nyhoff & Nyhoff, 2017).

Substring adalah karakter yang berurutan dalam *string*, masing-masing *substring* atau elemen penyusun *string* memiliki indeks. Elemen *string* yang memiliki index 0 dikenal dengan istilah prefix. Elemen *string* yang memiliki index terakhir dikenal dengan istilah suffix (Laaksonen, 2017). Salah satu bidang yang meneliti tentang kumpulan *string* atau text adalah text mining.

Text mining adalah disiplin ilmu yang menggabungkan ilmu bahasa dan ilmu komputer dengan teknik statistik dan pembelajaran mesin. Text mining digunakan dalam analisis teks dan mengubah teks dari bentuk yang tidak terstruktur menjadi bentuk yang lebih terstruktur Cielen, D., Meysman, A. D. B., & Ali, M. (2016).

BAB 14 STRUKTUR DATA *LIST*, *STACK* DAN *QUEUE*

Pendahuluan

Struktur data adalah cara atau metode untuk menyimpan dan mengatur data dalam komputer agar dapat digunakan secara efisien (Canning et al., 2023). Setiap jenis struktur data memiliki karakteristik dan kegunaan tertentu, tergantung pada bagaimana data diakses, disisipkan, atau dihapus. Beberapa struktur data dasar yang umum digunakan dalam pemrograman adalah *list* (daftar), *stack* (tumpukan), dan *queue* (antrian), yang masing-masing menyimpan kumpulan elemen dan menyediakan operasi tertentu untuk memanipulasinya.

Memahami struktur data sangat penting dalam pemrograman karena pilihan struktur data yang tepat dapat meningkatkan efisiensi dan performa program secara signifikan (Ren & Li, 2025). Selain itu, struktur data juga membantu *programmer* dalam menyelesaikan masalah secara sistematis, seperti pengelolaan memori, pencarian data, dan pengurutan. Dengan pemahaman yang baik, seorang *programmer* dapat menulis kode yang lebih bersih, terstruktur, dan mudah dikembangkan.

List, *stack*, dan *queue* memiliki perbedaan dalam cara mereka menyimpan dan mengakses data (Soni, 2024). *List* bersifat fleksibel dan memungkinkan akses langsung ke elemen mana pun berdasarkan indeks. *Stack* mengikuti prinsip LIFO (*Last In, First Out*), di mana elemen terakhir yang dimasukkan

BAB 15 ALGORITMA SORTING: BUBBLE SORT, SELECTION SORT, INSERTION SORT

Pendahuluan

Dalam ilmu komputer, proses pengurutan (*sorting*) data memiliki peran penting karena menjadi dasar bagi berbagai algoritma lain yang mengandalkan keterurutan data. Contoh nyata adalah algoritma *Binary Search*, yang memerlukan data dalam kondisi terurut (*sorted*) agar dapat beroperasi secara optimal. Di samping itu, pengurutan (*sorting*) juga memiliki penerapan luas dalam kehidupan sehari-hari, seperti dalam penyusunan daftar nama, nomor telepon, dan jenis data lain yang sering digunakan sehari-hari.

Pengurutan (*sorting*) merupakan metode untuk menyusun data yang awalnya tidak beraturan menjadi teratur berdasarkan kriteria atau aturan tertentu. Terdapat dua jenis metode pengurutan (*sorting*) data yang umum digunakan, yaitu:

1. *Ascending Order* (Pengurutan Menaik)

Pengurutan menaik merupakan proses penataan data dimulai dari nilai yang paling rendah hingga mencapai nilai yang paling tinggi.

2. *Descending Order* (Pengurutan Menurun)

Pengurutan menurun adalah proses pengaturan data dari nilai tertinggi ke nilai terendah.

Tiga algoritma pengurutan (*sorting*) yang paling mendasar, tetapi sangat penting adalah *Bubble Sort*, *Selection*

BAB 16 PENGENALAN REKURSI DALAM ALGORITMA

Pendahuluan

Dalam dunia pemrograman, terdapat berbagai pendekatan untuk menyelesaikan permasalahan komputasi. Salah satu pendekatan yang sangat penting dan sering digunakan adalah rekursi. Jika sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah secara langsung atau tidak langsung, ini disebut rekursi. Metode ini sangat efektif dalam menyelesaikan masalah yang dapat dipecah menjadi submasalah yang lebih kecil yang memiliki struktur yang sama dengan masalah awal.

Sebelum memahami lebih jauh bagaimana rekursi bekerja, penting untuk mengetahui bahwa teknik ini tidak hanya digunakan dalam contoh-contoh sederhana, tetapi juga diterapkan dalam berbagai bidang pemrograman. Misalnya, dalam matematika, rekursi digunakan untuk menghitung nilai faktorial suatu bilangan atau menghasilkan deret angka Fibonacci. Selain itu, dalam dunia pemrograman yang lebih kompleks, rekursi juga sangat berguna untuk memproses struktur data seperti pohon dan grafik. Struktur-struktur ini memiliki bentuk bercabang dan bertingkat, sehingga pendekatan rekursif sangat cocok digunakan karena mampu menelusuri data secara menyeluruh dengan cara yang efisien dan elegan (Fathansyah, 2007; Wahyono, 2016).

Namun demikian, penggunaan rekursi juga menuntut

BAB 17 ALGORITMA PENCARIAN: *LINEAR SEARCH* DAN *BINARY SEARCH*

Pendahuluan

Dalam bidang ilmu komputer, pencarian data merupakan salah satu proses fundamental yang sangat sering digunakan dalam berbagai aplikasi. Proses ini melibatkan usaha untuk menemukan posisi atau keberadaan suatu nilai tertentu di dalam sekumpulan data, seperti daftar angka, kumpulan nama, atau database.

Agar proses pencarian bisa dilakukan dengan efisien dan tidak memakan banyak waktu, dibutuhkan algoritma yang tepat. Pemilihan algoritma pencarian ini tergantung pada bagaimana struktur data disusun dan seberapa besar ukuran datanya.

Dua algoritma pencarian yang paling mendasar namun sangat penting adalah *Linear Search* dan *Binary Search*:

1. *Linear Search* bekerja dengan cara yang sederhana, yaitu memeriksa satu per satu setiap elemen dari awal hingga akhir hingga menemukan nilai yang dicari. Meskipun mudah diimplementasikan dan tidak memerlukan kondisi khusus pada data, algoritma ini menjadi kurang efisien ketika data berjumlah besar, karena harus melakukan pemeriksaan berulang.
2. *Binary Search*, di sisi lain, menggunakan pendekatan pembagian ruang pencarian secara bertahap. Dengan memanfaatkan fakta bahwa data telah diurutkan terlebih

BAB 18 PEMROGRAMAN MODULAR: FUNGSI DAN PROSEDUR DALAM ALGORITMA

Pendahuluan

Di era teknologi informasi yang terus berkembang, aplikasi perangkat lunak semakin kompleks dan beragam. Tantangan utama dalam pengembangan perangkat lunak modern adalah bagaimana menyusun program yang rapi, terstruktur, dan mudah diubah atau diperbaiki seiring waktu. Untuk menjawab tantangan tersebut, konsep pemrograman modular menjadi sangat relevan. Pendekatan ini memecah program besar menjadi bagian-bagian kecil atau modul, di mana setiap modul memiliki tanggung jawab spesifik. Dua elemen utama dalam modularisasi adalah fungsi dan prosedur.

1. Fungsi bertugas untuk memproses input dan mengembalikan hasil berupa nilai, sedangkan
2. Prosedur menjalankan serangkaian perintah tanpa mengembalikan nilai secara langsung.

Pendekatan modular ini membantu menjaga kejelasan struktur program, memudahkan pemeliharaan, dan mendukung pengembangan berkelanjutan, terutama dalam tim pengembang. Bab ini akan menguraikan konsep-konsep dasar tersebut secara mendalam dan memberikan banyak contoh praktis.

BAB 19 PEMROSESAN DATA DENGAN ALGORITMA GREEDY & *DIVIDE AND CONQUER*

Pendahuluan

Volume data global yang dihasilkan oleh berbagai sumber, mulai dari transaksi bisnis hingga sensor *Internet of Things* (IoT), telah meningkat secara eksponensial dalam beberapa tahun terakhir. International Data Corporation (IDC) memproyeksikan bahwa total data di seluruh dunia akan tumbuh dari 33 zettabyte pada tahun 2018 menjadi 175 zettabyte pada tahun 2025. Pertumbuhan masif ini menjadikan pemrosesan data yang efektif dan efisien sebagai kebutuhan yang semakin mendesak. Pemrosesan data tidak hanya mencakup penyimpanan, tetapi juga mencakup pembersihan, transformasi, analisis, dan penarikan informasi berharga dari himpunan data besar. Untuk mencapai hal tersebut, peran algoritma yang andal dan optimal menjadi krusial dalam memastikan data dapat diolah dengan cepat dan akurat.

Pemilihan metode algoritmik yang tepat berperan penting dalam mengatasi tantangan pemrosesan data skala besar. Algoritma yang kurang efisien dapat menyebabkan waktu eksekusi yang lama, penggunaan sumber daya yang berlebihan, atau bahkan kegagalan dalam menyelesaikan pemrosesan ketika data berukuran sangat besar. Oleh karena itu, pengembangan strategi algoritma yang cerdas menjadi salah satu fokus utama di bidang ilmu komputer dan rekayasa perangkat lunak. Dua pendekatan klasik yang menonjol dalam

BAB 20 PENGENALAN BAHASA PEMROGRAMAN DAN IMPLEMENTASI ALGORITMA

Pendahuluan

Setiap algoritma pada dasarnya hanyalah sebuah rencana logis—langkah demi langkah untuk menyelesaikan masalah tertentu. Namun, selama ide itu belum diwujudkan dalam bentuk kode yang dapat dijalankan, ia tetaplah abstrak. Bahasa pemrograman hadir sebagai jembatan utama antara logika dan eksekusi, antara ide dan solusi nyata.

Bab ini memperkenalkan bagaimana algoritma yang telah dipelajari sebelumnya dapat diterapkan secara langsung menggunakan Python. Dengan sintaks yang bersih dan mendekati *Pseudocode*, Python menjadi alat yang efisien untuk mempraktikkan berbagai algoritma, mulai dari pencarian, pengurutan, hingga pemrosesan data sederhana. Fokus utama bukan lagi pada konsep, tetapi pada bagaimana ide-ide tersebut diubah menjadi instruksi program yang berjalan.

Melalui praktik ini, pembaca diajak untuk tidak hanya memahami struktur kode, tetapi juga mulai membangun kebiasaan menulis program yang rapi, modular, dan mudah dikembangkan. Dengan fondasi ini, Anda akan lebih siap melangkah ke tahap berikutnya dalam dunia pemrograman—apakah itu membangun aplikasi, menganalisis data, atau mengembangkan solusi berbasis teknologi yang lebih kompleks.

Daftar Pustaka

- Abdillah, A. (2014). Inovasi dan Pengembangan Produk UKM Handikraf untuk Pasar Pariwisata di Bali. Program Studi Pariwisata, Jurusan Administrasi Bisnis FIA UB.
- Adiningsih, S. (2003). The Indonesia Business Rop in AFTA, Indonesia Business Perspective, Volume V, No. 3, PT. Harvest International Indonesia.
- Afiff, F. (2012). Pilar-Pilar Ekonomi Kreatif. Artikel. Universitas Bina Nusantara: Jakarta.
- Afrizal Zein & Chrisantus Trisianto. (2025). Algoritma dan Struktur Data Menggunakan Pemrograman Python. EUREKA MEDIA AKSARA.
- Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). Compilers: Principles, Techniques, and Tools (2nd ed.). Boston, MA: Addison Wesley.
- Alfina, Adi Ahmad, Yeni Yanti, & M. (2024). Algoritma dan Pemrograman. In Algoritma dan Pemrograman (Issue 1). CV. Naskah Aceh.
- Alotaibi, H., & Alotaibi, M. (2020). The effectiveness of *flowcharts* in enhancing algorithmic thinking among computer science students. International Journal of Emerging Technologies in Learning (iJET), 15(11), 4-18. <https://doi.org/10.3991/ijet.v15i11.13567>
- Barakbah, A. R., Karlita, T., & Ahsan, A. S. (2013). Logika dan Algoritma. Politeknik Elektronika Negeri Surabaya.
- Bhasin, H. (2023). Python Programming Using Problem Solving. Mercury Learning and Information.
- Bhasin, H., 2018. Python basics: a self-teaching Introduction. Mercury Learning and Information.
- Bokare, M. M., & Suryawanshi, A. (2024). C Language Foundation Textbook for BCA UG Course. Sankalp Publication.

- Caferra, R. (2011). *Logic for Computer Science and Artificial Intelligence*. John Wiley & Sons, Inc. ISBN 978-1-84821-301-2.
- Canning, J., Broder, A., & Lafore, R. (2023). *Data Structures & Algorithms in Python*.
- Chauhan, Y., & Duggal, A. (2020). Different *Sorting* algorithms comparison based upon the time complexity. *International Journal of Research and Analytical Reviews (IJRAR)*, 7(3), 114–118. https://www.researchgate.net/publication/344280789_Different_Sorting_Algorithms_comparison_based_upon_the_Time_Complexity
- Chen, Y., & Wang, L. (2021). *Pseudocode* as a pedagogical tool in introductory programming education. *Education and Information Technologies*, 26(3), 2873–2890. <https://doi.org/10.1007/s10639-020-10391-2>
- Cielen, D., Meysman, A. D. B., & Ali, M. (2016). *Introduction to data science*. Manning Shelter Island.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). Cambridge, MA: MIT Press.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.
- D'Souza, D., & Reddy, R. (2020). A beginner's approach to control structures in programming languages. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 6(3), 245–251.
- Dagiene, V., & Stupuriene, G. (2020). Bebras challenge and algorithmic thinking development. *Olympiads in Informatics*, 14, 21–34. <https://doi.org/10.15388/ioi.2020.03>
- Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. In *OSDI'04: Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, 137–150.

- Deitel, P. J., & Deitel, H. M. (2019). *C How to Program* (10th ed.). Pearson Education.
- Dicoding, 2025. *Academi: Memulai Pemrograman dengan Python*. <https://www.dicoding.com/academies/86/tutorials/4769>
- Downey, A. B. (2020). *Think Python: How to Think Like a Computer Scientist* (2nd ed.). O'Reilly Media.
- Elmqvist, N., & Irani, P. (2021). Visualizing algorithms: A survey of tools and techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(6), 2985-3001. <https://doi.org/10.1109/TVCG.2020.2975790>
- Endres, M., Weimer, W., & Kamil, A. (2021). An Analysis of Iterative and Recursive Problem Performance. *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 321-327. <https://doi.org/10.1145/3408877.3432391>
- Ennis, R. H. (2011). The nature of critical thinking: An outline of critical thinking dispositions and abilities. University of Illinois. Retrieved from http://faculty.education.illinois.edu/rhennis/documents/TheNatureofCriticalThinking_51711_000.pdf
- Fathansyah, R. (2007). *Struktur Data dan Algoritma dengan Bahasa C*. Bandung: *Informatika*.
- Forouzan, B. A. (2017). *Foundations of Computer Science* (4th ed.). Cengage Learning.
- Futschek, G., & Moschitz, J. (2022). Teaching computational thinking with *flowcharts* and *Pseudocode*: A classroom experiment. *ACM Transactions on Computing Education*, 22(4), 1-22. <https://doi.org/10.1145/3517131>
- Gellenbeck, E., & McConnell, J. (2020). Using *Pseudocode* to bridge the gap between problem-solving and coding. *Journal of Computing Sciences in Colleges*, 35(6), 92-100.
- Ginting, D. (2023). *Modul Perkuliahan: Mekanika Komputasi - Operator Logika dan Relasional*. Jakarta: Program Studi Magister Teknik Mesin, Universitas Mercu Buana.

- Ginting, D. (2023). Modul Perkuliahan: Mekanika Komputasi - Variable dan *Array*. Jakarta: Program Studi Magister Teknik Mesin, Universitas Mercu Buana.
- Goel, K., Dwivedi, P., & Sharma, O. (2023). Performance analysis of various *Sorting* algorithms: Comparison and optimization. 2023 11th International Conference on Intelligent Systems and Embedded Design (ISED), 1–5. <https://doi.org/10.1109/ISED56462.2023.10444609>
- Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2022). Data structures & algorithms in Java (6th ed.). Wiley.
- Goodrich, M. T., Tamassia, R., & Goldwasser, M. H., 2014. Data Structures and Algorithms in Java™. 6th Edition. Wiley.
- Govier, T. (2010). A practical study of argument (7th ed.). Wadsworth Cengage Learning.
- Grami, A. (2023). Algorithms. In Discrete Mathematics (pp. 211–230). Elsevier. <https://doi.org/10.1016/B978-0-12-820656-0.00012-5>
- Guerrini, V., & Pisanti, N. (2024). Algorithms Foundations. In Reference Module in Life Sciences. Elsevier. <https://doi.org/10.1016/B978-0-323-95502-7.00008-7>
- Halimatussyah'diyah Purba, & Yahfizham Yahfizham. (2023). Konsep Dasar Pemahaman Algoritma Pemrograman. Jurnal Arjuna: Publikasi Ilmu Pendidikan, Bahasa Dan Matematika, 1(6), 290–301. <https://doi.org/10.61132/arjuna.v1i6.356>
- Hazzan, O., & Lapidot, T. (2021). Guide to teaching computer science: An activity-based approach (2nd ed.). Springer. <https://doi.org/10.1007/978-3-030-39360-1>
- Herlambang, S. (2018). Implementasi Fungsi Rekursif Dalam Algoritma dan Perbandingannya dengan Fungsi Iteratif. 6. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2008-2009/Makalah2008/Makalah0809-079.pdf>
- Hidayat, A., Agustin, N., Misriati, T., Prayudani, S., Sibyan, H., Aryanti, R., Hidayat, R., Iswanto, M. E., Taufiqurrochman, M., Zein, A. A., Lase, Y. Y., Erlinda, S.,

- & Wilda, A. N. (2024). Struktur data menggunakan bahasa pemrograman C++ (Tundo, Ed.). Malang: Future Science.
- Hoare, C. A. R. (1962). *QuickSort*. *The Computer Journal*, 5(1), 10–16. (contoh algoritma *divide-and-conquer* untuk *Sorting*)
<https://www.geeksforgeeks.org/>
<https://www.khanacademy.org/computing/computer-science/algorithms>
<https://www.tutorialspoint.com/>
- Huffman, D. A. (1952). A Method *for* the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, 40(9), 1098-1101.
- Hurley, P. J. (2015). A concise introduction to logic (12th ed.). Cengage Learning.
- Izadkhan, H. (2022). *Problems on Algorithms: A Comprehensive Exercise Book for Students in Software Engineering*. Germany: Springer International Publishing.
- Jay Wengrow, 2017. “A Common-Sense Guide to Data Structures and Algorithms: Level Up Your Core Programming Skills”, Pragmatic Bookshelf; 1 edition.
- Jon Kleinberg, Éva Tardos, 2006. “Algorithm Design”, 1st Edition, Pearson; 1 edition (March 26, 2005).
- Kadir, A. (2012). *Algoritma & Pemrograman menggunakan C & C++* (1 ed., Vols. -). (B. R. W, Ed., & -, Trans.) Yogyakarta: ANDI.
- Kadir, A. (2012). *Algoritma & pemrograman menggunakan Java* (1 ed., Vols. -). (F. S. Suyantoro, Ed., & -, Trans.) Yogyakarta: ANDI.
- Kadir, A. (2020). *Dasar Logika Pemrograman Komputer* (Edisi Terbaru). Elex Media Komputindo.
- Karumanchi, N. (2016). *Data structures and algorithmic thinking with*
- Karumanchi, N., 2016. *Data Structure and Alorithmic Thinking with Python*. CareerMonk Publications.

- Karumanchi, N., 2017. *Data Structures And Algorithms Made Easy*. CareerMonk Publications.
- Katai, Z., & Toth, L. (2023). Enhancing programming skills with visual tools: The role of *flowcharts*. *Computers & Education*, 194, 104681. <https://doi.org/10.1016/j.compedu.2022.104681>
- Kleinberg, J., & Tardos, É. (2006). *Algorithm Design*. Addison-Wesley. (Greedy algorithms chapter)
- Knuth, D. E. (1997). *The Art of Computer Programming, Volume 1: Fundamental Algorithms* (3rd ed.). Reading, MA: Addison Wesley Longman.
- Knuth, D. E. (1998). *The Art of Computer Programming, Volume 3: Sorting and Searching* (2nd ed.). Addison-Wesley.
- Kumar, R., & Sharma, S. (2021). Analyzing the learning curve of programming control structures among beginners. *Journal of Computer Applications*, 44(2), 112–119.
- Laaksonen, A. (2017). *Guide to competitive programming: Learning and improving algorithms through contests*. Springer.
- Li, T. (2025). Research on the Application of Python in Big Data Analysis. *Journal of Artificial Intelligence and Information*, 2, 37–41.
- Liang, Y. D. (2018). *Introduction to Java Programming and Data Structures* (11th ed.). Pearson.
- Loukanova, R., Lumsdaine, P. L., & Muskens, R. (2021). *Logic and Algorithms in Computational Linguistics*. Springer. *Studies in Computational Intelligence*. ISBN 978-3-031-21779-1.
- Lutfina, E., & Ramadhan, F. L. (2019). Perbandingan Kinerja Metode Iteratif dan Metode Rekursif dalam Algoritma *Binary Search*. Seminar Nasional APTIKOM, 2019.
- Luxton-Reilly, A., & Denny, P. (2020). The impact of *Pseudocode* on novice *programmers'* understanding of algorithms. *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science*

- Education, 568-574.
<https://doi.org/10.1145/3341525.3387409>
- Malik, D. S. (2017). C++ Programming: From Problem Analysis to Program Design (8th ed.). Cengage Learning.
- Malik, D.S., 2018. “Pengantar Algoritma dan Pemrograman”, Boston: Cengage Learning.
- Malik, K. (2020). Logika dan Algoritma Pemrograman Komputer. Pustaka Nurja.
- Malmi, L., & Korhonen, A. (2021). Visual algorithm simulation: A review of tools and their educational impact. *ACM Transactions on Computing Education*, 21(2), 1-28.
<https://doi.org/10.1145/3445989>
- McCracken, M., & Waters, R. (2022). Data structures and algorithms using Python (3rd ed.). Pearson Education.
- Mirkowska, G., & Salwicki, A. (1987). Algorithmic Logic. D. Reidel Publishing Company. ISBN 90-277-1928-4.
- Mitchell, J. C. (2002). Concepts in Programming Languages. Cambridge University Press.
- Moore, B. N., & Parker, R. (2012). Critical thinking (10th ed.). McGraw-Hill Education.
- Munir Rinaldi, 2005. “Algoritma dan Pemrograman dalam Bahasa Pascal dan C”, *Informatika Bandung*.
- Nugroho Adi, 2009, “Algoritma dan struktur data dengan C#”, Pustaka ID, (1st edition).
- Nugroho, A., & Kurniawan, F. (2022). Optimasi pencarian pada basis data terdistribusi. *Jurnal Teknologi Informasi dan Komunikasi*, 10(1), 45-56.
<https://doi.org/10.1234/jtik.v10i1.12345>
- Nyhoff, J. L., & Nyhoff, L. R. (2017). Processing: An introduction to programming. CRC Press.
- Prasetya, R. A., & Wijaya, A. (2023). Penerapan algoritma *Binary Search* untuk pengindeksan dokumen elektronik. *Jurnal Informatika dan Sistem Informasi*, 9(3), 212-224.
<https://doi.org/10.5678/jisi.v9i3.67890>

- Prim, R. C. (1957). Shortest Connection Networks And Some Generalizations. *The Bell System Technical Journal*, 36(6), 1389–1401. (contoh algoritma greedy pada graf)
- Putri, M. P., Barovih, G., Azdy, R. A., Yuniansyah, Saputra, A., Sriyeni, Y., Rini, A., & Admojo, F. T. (2022). *Algoritma dan Struktur Data*. Bandung: Widina Bhakti Persada.
- Python. Careet Monk Publications.
- Raharjo, B., & Santoso, D. (2021). *Algoritma dan struktur data*. Andi Offset.
- Raharjo, B., & Santoso, D. (2021). *Algoritma dan Struktur Data*. Yogyakarta: Andi Offset.
- Ramalho, L. (2015). *Fluent Python*. Sebastopol, CA: O'Reilly Media.
- Ren, J., & Li, A. (2025). *Binary Search*. In *Silicon Valley Python Engineer Interview Guide: Data Structure, Algorithm, and System Design* (pp. 215–225). Springer.
- Ren, J., & Li, A. (2025). *Data Structure, Algorithm, and System Design*. Springer Nature Singapore. <https://doi.org/10.1007/978-981-96-3201-5>
- Rizvi, Q. M., Rai, H., & Jaiswal, R. (2024). *Sorting algorithms in focus: A critical examination of Sorting algorithm performance*. *International Conference on Emerging Trends in IoT & Computing Technologies-2023*, 1–5. https://www.researchgate.net/publication/378962637_Sorting_Algorithms_in_Focus_A_Critical_Examination_of_Sorting_Algorithm_Performance
- Robins, A., & Rountree, J. (2023). Learning to program: The role of scaffolding with *Pseudocode* and *flowcharts*. *Computer Science Education*, 33(1), 45-67. <https://doi.org/10.1080/08993408.2022.2034567>
- Rosa, A. S. (2018). *Logika Algoritma dan Pemrograman Dasar*. Modula.
- Sahu, U. (2024). Comparison among *selection Sort*, *Bubble Sort*, and *Insertion Sort*. Interview Kickstart Blog. Retrieved from

- <https://interviewkickstart.com/blogs/learn/comparison-among-bubble-Sort-selection-Sort-and-insertion-Sort>
- Salmon, M. H. (2012). Introduction to logic and critical thinking (6th ed.). Cengage Learning.
- Schäfer, A. (2023). Efficient algorithms for big data analytics. Springer.
- Sebesta, R. W. (2018). Concepts of Programming Languages (12th ed.). Pearson.
- Sebesta, R. W. (2019). Concepts of Programming Languages (12th ed.). Pearson.
- Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Boston, MA: Addison-Wesley Professional.
- Sedgewick, R., & Wayne, K. (2020). Algorithms (4th ed.). Addison-Wesley.
- Sedgewick, R., & Wayne, K. (2022). Algorithms (5th ed.). Addison-Wesley Professional.
- Sharma, P. B. (2025). Principles Of Programming Language Paradigms. OrangeBooks Publication.
- Soni, M. (2024). Mastering Algorithms & Data Structures.
- Steven S S. Skiena, 2010. "The Algorithm Design Manual 2nd" Springer;
- Sweigart, A. (2025). Automate the Boring Stuff with Python. No Starch Press.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 2022. "Introduction to Algorithms", The MIT Press, (5th Edition).
- Vierdansyah, D., Sihombing, R., & Damanik, M. (2023). Perbandingan Analisis *Bubble Sort* dan *Insertion Sort* pada Pengurutan Data Sewa Kios. COELITE: Journal of Computer Engineering, Electronics and Information Technology, 2(1), 1–6. <https://doi.org/10.17509/coelite.v2i1.57091>
- Vihavainen, A., & Vikberg, T. (2020). Algorithm visualization and its effect on learning outcomes in programming courses. Journal of Educational Technology Systems,

- 48(4), 492-510.
<https://doi.org/10.1177/0047239520904701>
- Vishwakarma, A., Agrwal, H., Thakur, S. S., Bairagi, T., Dahariya, Y., Mahilang, H., Gohil, C., & Kumar, P. K. (2024). Recursion Algorithms: Principles, Applications, And Innovations. 04, 8263–8270.
- Wahyono, T. (2016). Struktur Data: Konsep dan Implementasi dalam Bahasa C/C++. Yogyakarta: Andi.
- Weiss, M. A., Data structures and algorithm analysis in C++, 4th Edition. Pearson.
- Wengrow, J., 2020. A Common-Sense Guide to Data Structures and Algorithms. 2nd Edition. The Pragmatic Programmers, LLC.
- Wirth, N. (1976). Algorithms + Data Structures = Programs. Englewood Cliffs, NJ: Prentice-Hall.
- Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). Algorithms. In Data Mining (pp. 91–160). Elsevier. <https://doi.org/10.1016/B978-0-12-804291-5.00004-0>
- Xtracts, C. (2023). PHP Basics - A Brief Guide. Mocktime Publication.
- Yanti, F., & Eriana, E. S. (2024). Algoritma *Sorting* dan searching. Purbalingga: CV. Eureka Media Aksara.
- Zelle, J. M. (2016). Python Programming: An Introduction to Computer Science (3rd ed.). Franklin, Beedle & Associates.
- Zhang, Y., & Li, X. (2021). A comparative analysis of search algorithms: Linear vs *Binary Search* in large-scale data. Journal of Computer Science, 15(2), 105–118. <https://doi.org/10.1016/j.jcs.2021.01.005>

Tentang Penulis



M. Rhifky Wayahdi, lahir di Medan, 05 Februari 1993, merupakan anak pertama dari tiga bersaudara. Penulis merupakan alumni Program Sarjana (S-1) di Universitas Potensi Utama pada Jurusan Sistem Informasi dan lulus tahun 2015. Penulis melanjutkan studi Program Magister (S-2) Teknik Informatika di Universitas Sumatera Utara dan lulus tahun 2019. Kemudian saat ini Penulis sedang melanjutkan Pendidikan Doktor (S-3) Ilmu Komputer di Universitas Sumatera Utara mulai 2024 sampai sekarang (on-going). Berkarir sebagai dosen dimulai dari tahun 2020 di Universitas Battuta. Mulai tahun 2023-sekarang diberi kepercayaan menjadi Dekan Fakultas Teknologi di Universitas Battuta.



Phie Chyan, ST, M.Cs, Lahir di Makassar 13 April 1981, Setelah menyelesaikan pendidikan menengah di SMA Katholik Cendrawasih Makassar Tahun 1996. Penulis melanjutkan pendidikan tinggi S1 di Universitas Atma Jaya Makassar program studi Teknik Elektro Kemudian S2 di Universitas Gadjah Mada Program Studi Ilmu Komputer dan Doktor (S3) di Universitas Hasannudin di Program Studi Elektro konsentrasi Teknik Informatika. Buku ini merupakan salah satu karya dari penulis sesuai bidang minat dan ilmu dari penulis dengan tujuan untuk berbagi ilmu pengetahuan kepada masyarakat.



Agung Yuliyanto Nugroho, M.Kom., M.Par Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2015 silam. Hal tersebut membuat penulis melanjutkan pendidikan ke Perguruan Tinggi dan berhasil menyelesaikan studi S1 di prodi Teknik Informatika Universitas Teknologi Yogyakarta pada tahun 2018. Dua tahun kemudian, penulis menyelesaikan studi S2 di prodi Teknik Informatika Program Pasca Sarjana Universitas Amikom Yogyakarta dan juga prodi Magister Pariwisata di Sekolah Tinggi Pariwisata Ambarrukmo Yogyakarta. Atas dedikasi dan kerja keras dalam membuat suatu karya, Republik Indonesia Kementerian Hukum Dan Hak Asasi Manusia sudah mencatat ada kurang lebih 100 karya yang sudah tercatat di surat pencatatan ciptaan sebagai salah satu kontribusi dalam melindungi hak kekayaan intelektual.



Dr. Amril Mutoi Siregar, M. Kom. lahir di Ujung Padang, Sumatera Utara. Sekolah SD sampai SMP diselesaikan di kota kelahirannya. Kemudian pada tahun 1994 melanjutkan Pendidikan di SMA PGRI 4 Jakarta Jurusan IPA. Tahun 2004 Melanjutkan Pendidikan Program S1 di STMIK MIC Cikarang pada jurusan Teknik Informatika. Tahun 2014 melanjutkan Pendidikan Program S2 di President University Cikarang Jurusan Teknik Informatika. Tahun 2020 melanjutkan Pendidikan Program S3 di IPB university jurusan Ilmu Komputer. Penulis konsentrasi mengajar dan penelitian di bidang: Machine Learning, Deep Learning, data mining, Data Science dan Algoritma.



Dr. Ir. Hansi Effendi, S.T., M.Kom., adalah seorang akademisi dan peneliti di bidang pendidikan teknologi dan ilmu komputer. Ia menyelesaikan pendidikan S1 di bidang Teknik Elektro di Universitas Andalas, kemudian meraih gelar S2 dalam Magister Ilmu Komputer di UPI YPTK Padang. Untuk memperdalam keahliannya, ia menyelesaikan studi S3 di Program Doktor Pendidikan Teknologi dan Kejuruan, Universitas Negeri Yogyakarta. Dengan pengalaman lebih dari 20 tahun di dunia pendidikan, Hansi Effendi telah berkontribusi sebagai dosen di Universitas Negeri Padang sejak tahun 2002. Fokus penelitiannya meliputi pengembangan metode pembelajaran inovatif, dengan penekanan pada pendekatan seperti Pembelajaran Berbasis Masalah (PBL) dan Pembelajaran Berbasis Proyek (PjBL). Ia juga aktif terlibat dalam berbagai proyek penelitian yang bertujuan meningkatkan kualitas pendidikan melalui pendekatan berbasis teknologi.



Satriawaty Mallu, Dosen di STMIK Profesional Makassar, menyelesaikan S1 Teknik Informatika Universitas Handayani, S2 Ilmu Komputer Universitas Gadjah Mada dan Insinyur Profesi UMI Makassar, sebagai Dosen salah satu kewajiban kita adalah menulis, buku ini adalah salah satu karya dan Insya Allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Samsuriah, S.Kom., M.T, buku ini adalah salah satu karya dan Insha Allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan. Lulus S1 di STMIK Handayani Makassar tahun 2012, kemudian menyelesaikan pendidikan S2 di program studi Teknik Elektro dengan konsentrasi Teknik Informatika pada Fakultas Teknik Universitas Hasanuddin Makassar pada tahun 2015. Pada tahun 2015 Menjadi Dosen di Prodi DIII Manajemen Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Profesional Makassar hingga saat ini, kemudian desember 2024 pindah di Jurusan/Prodi Ilmu Komputer sekaligus diberikan amanah sebagai Koordinator Program Studi di Program Studi Ilmu Komputer. Penulis mengajarkan beberapa mata kuliah seperti, Sistem Operasi, Teknik Riset Operasi, Pemrograman Berorientasi Objek, dll. Aktif melaksanakan penelitian terutama yang berhubungan dengan bidang ilmu yang di tekuni yaitu komputer serta melakukan kerja sama penelitian dengan berbagai bidang ilmu yang lain. Selain itu penulis telah menerbitkan artikel pada berbagai jurnal nasional.



Dr. Dianta Ginting, EMBA. lahir di desa Kutatengah, Kabanjahe, Kabupaten Karo, Sumatera Utara. Pendidikan dasar (SD) diselesaikan di desa kelahirannya, kemudian melanjutkan ke SMP Negeri 1 Kabanjahe. Pada tahun 1999, beliau menyelesaikan pendidikan menengah atas di SMA Negeri 2 Kabanjahe. Tahun 2002, Dr. Dianta Ginting melanjutkan pendidikan S1 di

bidang Geofisika di Institut Teknologi Bandung (ITB). Setelah menyelesaikan studi sarjana, beliau melanjutkan ke jenjang magister (S2) pada tahun 2009 di Chungbuk National University, Korea Selatan. Ketertarikan dalam pengembangan keilmuan membawanya kembali melanjutkan studi doctoral (S3) pada tahun 2015 di Kyung Hee University, Korea Selatan.



Ni Putu Linda Santiari, S.Kom., M.Kom.
Lahir di Badung, 07 Oktober 1990. Lulus S2 di Program Studi Magister Teknik *Informatika* Universitas Amikom Yogyakarta tahun 2016. Saat ini sebagai Dosen di Institut Teknologi dan Bisnis STIKOM Bali pada Fakultas *Informatika* dan Komputer.



Ismi Rizqa Lina, S.Mat., M.at merupakan seorang Dosen prodi Sains Data Universitas Insan Cita Indonesia. Ia menyelesaikan Pendidikan S1 Matematika di UIN Maulana Malik Ibrahim Malang tahun 2027, kemudian melanjutkan Pendidikan S2 Matematika di Universitas Brawijaya tahun 2020. Buku *Algoritma dan Pemrograman* merupakan buku ke 2 yang ditulis setelah menulis buku *Pengantar Statistik*. Selain itu ia juga fokus penelitian di bidang matematika dan Machine Learning.



Dr. Drs. Waris Marsisno, M.Stat. merupakan dosen tetap pada Program Studi Komputasi Statistik, Politeknik Statistika STIS. Bagian yang ditulis dalam buku ini merupakan salah satu karya dan semoga secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Fahmi Ruziq, lahir di Banda Aceh, 16 Juni 1989, merupakan anak pertama dari empat bersaudara. Penulis merupakan alumni Program Sarjana (S-1) di Universitas Serambi Mekkah pada Jurusan Teknik *Informatika* dan lulus tahun 2010. Penulis melanjutkan studi Program Magister (S-2) Teknik *Informatika* di Universitas Sumatera Utara dan lulus tahun 2020. Berkarir sebagai dosen dimulai dari tahun 2020 di Universitas Battuta. Mulai tahun 2022-sekarang diberi kepercayaan menjadi Ketua Program Studi Sistem Informasi Fakultas Teknologi di Universitas Battuta.



Mardiah, S.SI., M.Kom Lahir di Bandung 6 November 1993, penulis menempuh pendidikan Sarjana di bidang Sistem Informasi di Universitas Sriwijaya, kemudian melanjutkan studi Magister Komputer di Institut Pertanian Bogor. Buku ini adalah salah satu karya dan Insya Allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis bertujuan untuk meneruskan ilmu yang dititipkan dan

salah satu bentuk usaha mendapatkan amal jariyah dari Allah. Barakallah fikum.



Anis Fitri Nur Masruriyah, adalah seorang pendidik dan peneliti yang aktif di dunia pendidikan tinggi di Indonesia. Ia lahir pada tahun yang tidak disebutkan dan telah aktif mengajar di salah satu universitas di Indonesia sejak tahun 2019. Penulis adalah seorang lulusan dari Institut Pertanian Bogor (IPB) meraih gelar M.Kom pada tahun 2018. Sebelumnya, pada tahun 2015 memperoleh gelar S.Kom dari Universitas Brawijaya. Selama karirnya di dunia pendidikan, Penulis telah aktif terlibat dalam penelitian dan pengabdian kepada masyarakat sejak tahun 2019. Penulis juga telah berkontribusi dalam upaya pengabdian masyarakat untuk memanfaatkan pengetahuannya demi kemajuan masyarakat dan negara. Sebagai seorang pendidik dan peneliti, buku ini adalah salah satu karya dan inshaa allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Santi Prayudani, lahir di Kotamadya Binjai Sumatera Utara pada tanggal 28 Maret 1986. Penulis menempuh pendidikan S1 di Prodi Ilmu Komputer Universitas Sumatera Utara pada tahun 2004. Kemudian melanjutkan kembali pendidikan S2 di Prodi Teknik Informatika Universitas Sumatera Utara pada tahun 2011. Memulai karir sebagai guru di SDS Al Azhar Medan pada tahun 2010. Kemudian mengajar juga sebagai dosen di AMIK

Harapan dan Universitas Pembangunan Panca Budi dari tahun 2011 sampai tahun 2014. Saat ini penulis diberi amanah oleh negara untuk mengabdikan sebagai dosen di Politeknik Negeri Medan dari tahun 2015.



Sri Retnowati, S.Pd., M.Pd. Lahir di Kebumen, 22 September 1994. Lulus S2 di Program Studi Pendidikan Matematika Universitas Sebelas Maret tahun 2020. Saat ini sebagai Dosen di Universitas Insan Cita Indonesia Program Studi Sains Data.



Ari Widiastono, S.Kom., MT.,MH, buku ini adalah salah satu karya dan inshaa allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Dita Novita Sari, S.Kom., M.T.I. Lahir di Gadingrejo, 06 November 1988. Lulus S2 di Program Teknik *Informatika* Institut *Informatika* dan Bisnis Darmajaya Lampung tahun 2015. Saat ini sebagai Dosen di Institut Bakti Nusantara Lampung Program Studi Sistem Informasi.



A.Kachsyfur Djasim Ilyas Paenrongi, buku ini adalah salah satu karya dan inshaa allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Triana Dewi Salma, M.T., Penulis lahir di Tuban pada tanggal 2 September 1994. Saat ini, penulis menjabat sebagai Dekan Fakultas Sains dan Bisnis serta dosen tetap pada Program Studi *Informatika* di Universitas LIA. Penulis memperoleh gelar Sarjana pada bidang *Informatika* dari Universitas Muhammadiyah Surakarta dan melanjutkan pendidikan Magister *Informatika* di Institut Teknologi Bandung. Dorongan utama dalam berkarya adalah keyakinan bahwa ilmu pengetahuan harus terus berkembang dan memberi manfaat jangka panjang. Buku ini adalah salah satu karya dan inshaa allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Buku Logika dan Algoritma Pemrograman: Fondasi Dasar Menjadi Programmer Andar disusun untuk membantu pembaca membangun pemahaman yang kuat terhadap konsep logika dan algoritma, dua elemen krusial dalam dunia pemrograman. Melalui pendekatan sistematis dan progresif, buku ini membekali pembaca dengan kemampuan berpikir logis, menyusun langkah-langkah penyelesaian masalah secara terstruktur, serta mengimplementasikannya ke dalam kode program. Isi buku mencakup berbagai topik penting mulai dari proposisi dan logika predikat, struktur kontrol algoritma, tipe data, operasi aritmetika dan logika, hingga konsep modularitas, rekursi, dan algoritma pemrosesan data seperti greedy dan divide-and-conquer. Disertai dengan contoh-contoh pseudocode, flowchart, dan implementasi nyata dalam bahasa Python, buku ini cocok digunakan baik sebagai bahan ajar di kelas maupun panduan belajar mandiri. Dirancang untuk pemula hingga menengah, buku ini menjadi referensi tepat bagi pelajar, mahasiswa, dosen, atau siapa pun yang ingin memulai karier sebagai programmer profesional. Dengan memahami fondasi logika dan algoritma secara menyeluruh, pembaca akan lebih siap menghadapi tantangan dunia pemrograman yang menuntut efisiensi, ketepatan, dan kreativitas dalam menyelesaikan masalah digital masa kini.

DITERBITKAN OLEH
PT. MIFANDI MANDIRI DIGITAL



Jln Payanibung Ujung D
Dalu Sepuluh-B, Tanjung Morawa
Kab. Deli Serdang Sumatera Utara

