LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
TAHUN AKADEMIK 2025/2026

Disusun Oleh :

Nama : Mochammad Ferdyansyah
NIM : 251080200060
Kelompok : 6

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO

LEMBAR PERSETUJUAN

Telah Diperiksa Dan Disetujui
Isi Laporan Ini

LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA

Disusun Oleh:

Nama : Mochammad Ferdyansyah
NIM : 251080200060
Kelompok : 6

Mengetahui,
Laboran Informatika

(**Melina Atikawati, S.Kom.**)

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025-2026

KATA PENGANTAR

Alhamdulillah segala puji syukur atas kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penyusun dapat menyelesaikan Laporan Praktikum Algoritma dan Pemrograman ini tanpa halangan yang berarti.

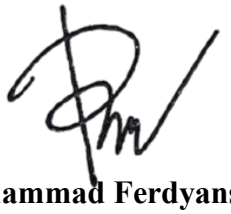
Keberhasilan penyusun dalam menyusun Laporan Praktikum Algoritma dan Pemrograman ini tidak lepas dari bantuan berbagai pihak. Untuk itu saya selaku penyusun menyampaikan terimakasih yang sebesar-besarnya kepada :

1. **Ir. Iswanto, S.T., M.MT.,** selaku Dekan Fakultas Sains dan Teknologi Universitas Muhammadiyah Sidoarjo.
2. **Ade Eviyanti, S.Kom., M.Kom.,** selaku Kepala Program Studi Informatika Universitas Muhammadiyah Sidoarjo.
3. **Uce Indahyanti, S.Kom., M.Kom.,** selaku Dosen Praktikum Algoritma dan Pemrograman Universitas Muhammadiyah Sidoarjo.
4. **Melina Atikawati, S.Kom.,** selaku Laboran Informatika Universitas Muhammadiyah Sidoarjo.
5. **Nadya Mulya Islami,** selaku Asisten praktikum yang telah membantu terlaksananya Praktikum Algoritma dan Pemrograman.
6. Keluarga dan teman-teman yang telah memberi bantuan baik materi maupun spiritual dalam pembuatan Laporan Algoritma dan Pemrograman

Saya selaku penyusun menyadari bahwa laporan ini masih jauh dari kesempurnaan, untuk itu sangat mengharapkan kritik dan saran dari berbagai pihak yang sifatnya membangun.

Akhir kata, semoga laporan ini dapat menjadi referensi untuk menambah wawasan para pembaca dan tentunya dapat memberikan manfaat sebagaimana yang diharapkan.

Sidoarjo, 05 April 2026



(Mochammad Ferdyansyah)

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

DAFTAR ISI

KATA PENGANTAR ..... **I**
DAFTAR ISI ..... **II**
BAB I PENDAHULUAN **1**
A. PROFIL LABORATORIUM 1
B. MANAJEMEN LABORATORIUM 2
C. SISTEM PENGELOLAAN 5
D. PROSEDUR DAN MEKANISME 6
E. TATA TERTIB 9
F. PENGGUNAAN LABORATORIUM 10
BAB II KURIKULUM **12**
A. PENDAHULUAN MATAKULIAH 12
B. TUJUAN PEMBELAJARAN 12
C. PRASYARAT MATAKULIAH 12
D. SILABUS PERKULIAHAN 13
E. DIAGRAM PENCAPAIAN KOMPETENSI 13
F. RENCANA PELAKSANAAN PEMBELAJARAN 14
BAB III **15**
POKOK BAHASAN 1 ..... **15**
A. PENDAHULUAN 15
B. TUJUAN 15
C. PENYAJIAN (TUTORIAL) 15
LEMBAR KERJA DAN TUGAS 20
POKOK BAHASAN 2 ..... **29**
A. PENDAHULUAN 29
B. TUJUAN 29
C. PENYAJIAN (TUTORIAL) 29
LEMBAR KERJA DAN TUGAS 32
POKOK BAHASAN 3 ..... **48**

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

A. PENDAHULUAN 48
B. TUJUAN 48
C. PENYAJIAN (TUTORIAL) 48
LEMBAR KERJA DAN TUGAS 54
POKOK BAHASAN 4 65
A. PENDAHULUAN 65
B. TUJUAN 65
C. PENYAJIAN (TUTORIAL) 65
LEMBAR KERJA DAN TUGAS 69
POKOK BAHASAN 5 82
A. PENDAHULUAN 82
B. TUJUAN 82
C. PENYAJIAN (TUTORIAL) 82
LEMBAR KERJA DAN TUGAS 84
POKOK BAHASAN 6 96
A. PENDAHULUAN 96
B. TUJUAN 96
C. PENYAJIAN (TUTORIAL) 96
LEMBAR KERJA DAN TUGAS 102
DAFTAR PUSTAKA 115

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

BAB I
PENDAHULUAN

A. PROFIL LABORATORIUM

Laboratorium Terpadu Fakultas Sains dan Teknologi digunakan sebagai pusat pembelajaran secara praktek dan eksperimental. Mahasiswa diharapkan akan dapat menerapkan materi kuliah secara langsung pada alat yang telah disediakan, mempelajari alat secara langsung, melakukan pengambilan data, penelitian, dan konsultasi. Dalam laboratorium ini terdapat dua 7 Program Studi yaitu : Informatika, Teknik Mesin, Teknik Industri, Teknik Elektro, Teknik Sipil, Agroteknologi, Teknologi Pangan.

1. **Visi Laboratorium** :

   Menjadi laboratorium terpadu yang unggul dan inovatif dalam pengembangan sains dan teknologi berdasarkan nilai-nilai Islam untuk kesejahteraan masyarakat tingkat ASEAN pada tahun 2038.

2. **Misi Laboratorium** :
   1) Melaksanakan kegiatan praktikum sebagai penerapan teori yang didapat selama perkuliahan.
   2) Menyediakan sarana dan prasarana untuk kegiatan penelitian dalam bidang rekayasa IPTEKS yang mendukung proses pembelajaran untuk kesejahteraan masyarakat.
   3) Menyediakan sarana dan prasarana untuk kegiatan pengabdian kepada masyarakat dalam rekayasa IPTEKS berbasis potensi lokal untuk kesejahteraan masyarakat.
   4) Memberikan layanan kerjasama dengan instansi pemerintah/swasta/ baik dalam maupun luar negeri yang berkelanjutan untuk menguatkan Catur Dharma Perguruan Tinggi Muhammadiyah (pendidikan pengajaran, penelitian, pengabdian dan Al Islam Kemuhammadiyahan).

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

5) Memberikan layanan pengujian, analisis, dan sertifikasi yang dapat dipertanggungjawabkan keakuratannya dan keabsahany

B. MANAJEMEN LABORATORIUM

1. Definisi

[1] Praktikum Algoritma dan Struktur Data adalah kegiatan akademis yang wajib dilakukan oleh setiap Mahasiswa Program Studi Informatika yang telah menempuh matakuliah prasarat Algoritma dan Struktur Data. Praktikum ini berbentuk praktek coding yang menekankan aspek kognitif atau berbentuk magang kerja dilaboratorium yang menekankan aspek psikomotorik diikuti aspek kognitif.

[2] Ka.Prodi adalah penanggung jawab secara administratif segala aktifitas yang ada di lingkungan Program Studi Informatika

[3] Ka.Lab adalah penanggung jawab secara administratif segala aktifitas yang ada di laboratorium terpadu fakultas sains dan teknologi.

[4] Laboran adalah wakil Ka.Lab dalam melaksanakan aktifitas sehari-hari yang ada dilingkungan masing-masing laboratorium program studi.

[5] Dosen pembimbing praktikum tim staf pengajar yang dibentuk oleh Program Studi yang bertugas membuat modul praktikum, menguji mahasiswa dan membimbing mahasiswa dalam penulisan laporan praktikum secara menyeluruh.

[6] Monitoring merupakan pemantauan yang dilakukan oleh Dosen pembimbing praktikum dan Ka.Lab untuk mengecek pelaksanaan praktikum oleh mahasiswa. Monitoring dilakukan secara acak dan minimum sekali dilakukan dalam satu periode pelaksanaan praktikum.

[7] *Logsheet* adalah lembar kerja yang berisi aktivitas kerja mahasiswa yang diketahui dan ditandatangani oleh asisten

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

praktikum gambar teknik.

[8] Ujian pretest dan posttest praktikum adalah ujian yang dilaksanakan sebelum dan sesudah pelaksanaan praktikum yang dilakukan oleh asisten laboratorium dan dosen pembimbing untuk memperoleh nilai praktikum.

[9] Laporan praktikum adalah naskah laporan praktikum yang ditulis sesuai dengan panduan penulisan praktikum dilaboratorium yang sudah direvisi setelah ujian dan melaporkan aktivitas pelaksanaan praktikum di laboratorium. Laporan praktikum dibuat dalam bentuk *hardcopy* dan *softcopy*

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

2. Waktu Dan Tempat Pelaksanaan

Waktu praktikum sesuai dengan jadwal yang telah ditentukan. Tempat pelaksanaan praktikum di Laboratorium yang sudah tercantum pada jadwal.

[1] Jumlah tatap muka

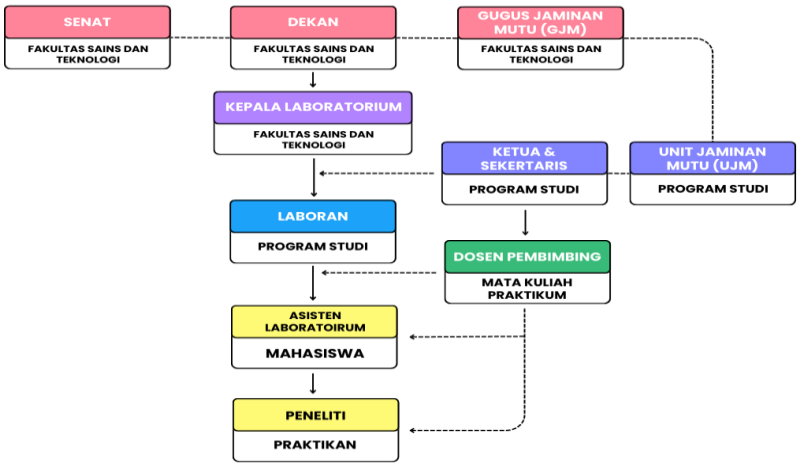
Jumlah tatap muka ditentukan dengan jumlah kelompok mahasiswa praktikum dengan pertimbangan memaksimalkan penggunaan peralatan praktikum, kapasitas laboratorium mampu menampung maksimal 40 mahasiswa praktikum.

[2] Lama praktikum setiap tatap muka

Lama praktikum setiap kelompok adalah 16 jam tatap muka ini dibagi dalam dua sesi, sesi hari pertama 8 jam tatap muka istirahat 1 jam kemudian sesi hari kedua 8 jam tatap muka istirahat 1 jam.

3. Struktur Organisasi

Laboratorium Terpadu berada di bawah struktur Fakultas Sains dan Teknologi Universitas Muhammadiyah Sidoarjo, seperti pada Gambar 1.1 berikut :



Gambar 1.1 Bagan Laboratorium Terpadu Fakultas Sains dan Teknologi Keterangan :

———— : Garis Komando
---------- : Garis Koordinasi

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

C. SISTEM PENGELOLAAN

1. Tugas pihak terkait :

[1] Praktikan

- Mendaftar sebagai calon praktikan
- Mengikuti praktikum sesuai jadwal
- Konsultasi kepada asisten dan dosen pembimbing
- Membuat laporan praktikum
- Mengikuti ujian akhir praktikum

[2] Asisten Praktikum

- Menyiapkan segala perlengkapan praktikum
- Menjelaskan modul praktikum
- Membimbing perbaikan laporan praktikum
- Membimbing praktikan membuat laporan
- Memberikan nilai praktikum
- Perawatan laboratorium

[3] Laboran

- Membuat daftar peserta praktikum
- Menyusun jadwal praktikum
- Melakukan pengecekan dan perawatan segala perlengkapan di laboratorium
- Mengevaluasi kegiatan praktikum

[4] Dosen Pembimbing

- Mendampingi kegiatan praktikum
- Membimbing perbaikan laporan praktikum
- Memberkan penilaian hasil praktikum

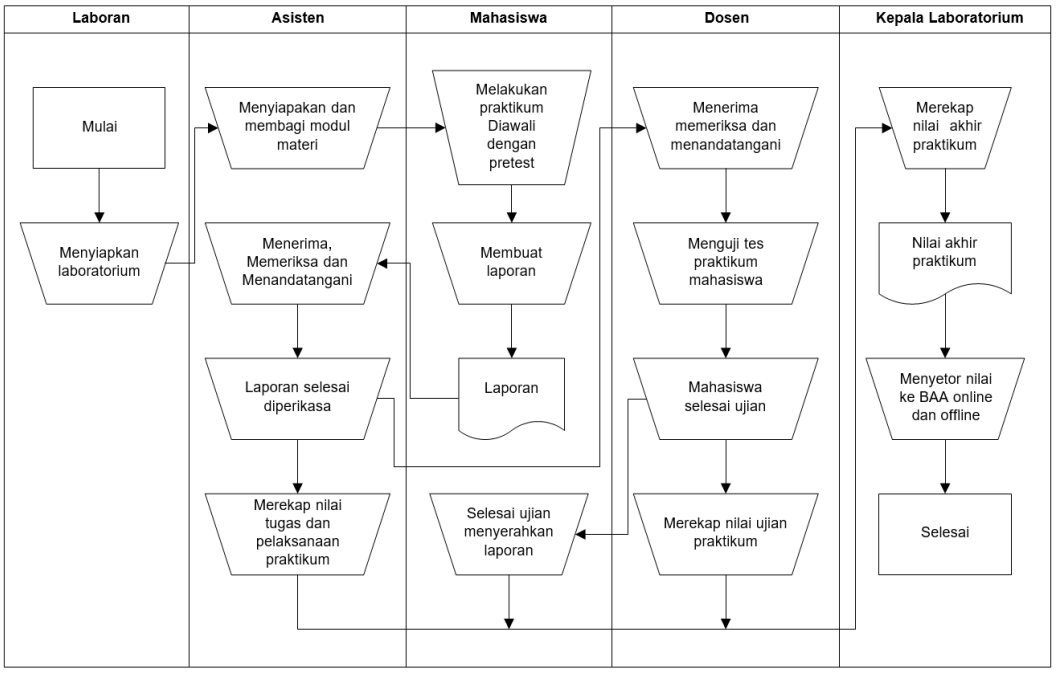
[5] Kepala Laboratorium

- Menkoordinasikan laboran dan asisten praktikum
- Mengajukan penambahan alat praktikum
- Membuat program kerja tahunan
- Menjalin kerjasama antar laboratorium internal maupun external

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

D. PROSEDUR DAN MEKANISME

Langkah-langkah yang dilakukan oleh Laboran, Asisten, Mahasiswa, Dosen dan Ka.lab disajikan dalam diagram alir pada Gambar 1.2.



Gambar 1.2 Diagram alir Langkah-langkah yang Dilakukan oleh Kepala Laboratorium, Mahasiswa, Asisten, Laboran dan Dosen Praktikum pada pelaksanaan praktikum.

Penjelasan :

1. Laboran menyiapkan laboratorium dan perlengkapannya, peralatan alat-alat kelengkapan praktikum yang dipimjam praktikan ditulis dalam form pemimjaman alat yang ditandatangani asisten dan perwakilan praktikan sebagai penanggung jawab peralatan.
2. Asisten menyiapkan materi praktikum berdasarkan modul praktikum
3. Mahasiswa melaksanakan pretest terlebih dahulu kemudian dilanjutkan dengan praktikum didampingi asisten, di monitoring dosen pembimbing dan Ka.Lab secara acak.
4. Mahasiswa membuat *Logsheet* dan diserahkan kepada asisten pada pertemuan berikutnya.
5. Mahasiswa wajib absen untuk setiap sesi praktikum
6. yang dilakukan oleh asisten praktikum.
7. Asisten memeriksa dan menandatangai asistensi laporan

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

praktikum mahasiswa, yang point koreksinya sesuai dengan petunjuk penulisan laporan praktikum kemudian yang menjadi point utama dalam koreksi diantaranya:

- Unsur originallitas.
- Kesesuain materi laporan dengan job sheet yang dikerjakan.
- Kedalaman analisis data pengukuran produk/quality control.
- Sistematika laporansesuai dengan petunjuk penulisan laporan praktikum.

8. Laporan yang sudah dikoreksi/diperiksa asisten dikembalikan lagi kepada mahasiswa, untuk menjadi pertimbangan penulisan laporan yang lebih baik sebelum maju asistensi ke dosen pembimbing.
9. Pada akhir praktikum, Dosen memberikan tes/ujian yang harus diikuti oleh semua mahasiswa praktikum.
10. Dosen menyerahkan nilai hasil tes/ujian praktikum yang dilaksanakan mahasiswa ke Kepala Laboratorium.
11. Asisten merekap nilai praktikum [10%P (tugas Pralab) + 10%K (Kehadiran) + 10%AS (Asistensi) + 10%LS (Laporan Sementara) + 30%LP (Laporan Praktikum) + 30%SH (Seminar Hasil / Ujian Praktikum)], kemudian Asisten menyerahkan nilai ke Laboran.
12. Mata Kuliah Praktikum

    Mata kuliah praktikum merupakan mata kuliah untuk mencapai capaian pembelajaran keterampilan khusus sekaligus pengetahuan. Penilaian untuk mata kuliah Praktikum meliputi aspek :

    a) Pralab (P)
    b) Kehadiran (K)
    c) Asisten (As)
    d) Laporan Sementara (LS)
    e) Laporan Praktikum (LP)
    f) Seminar Hasil (SH)

Sehingga dapat dirumuskan penilaian KKN sebagai berikut:

13. Laboran merekap nilai praktikum dan menyerakhan nilai ke dosen pengampu yang nantinya nilai praktikum dimaukkan

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

nilau tugas dari kompilasi [5%K (Kehadiran) + 5%P (Aktifitas Partisipasi Aktif) + 20%Pro (Hasisl Proyek) + 10%Q (Quiz) + 20%T (Tugas) + 20%UTS (Ujian Tengah Smester + 20%UAS (Ujian Akhir Semester)].

$$NA = \frac{1 \times P + 1 \times K + 1 \times As + 1 \times LS + 3 \times LP + 3 \times SH}{10}$$

Keterangan:

- Kehadiran (K)
- Aktivitas Partisipasi Aktif (P)
- Hasil Proyek (Pro)
- Quiz (Q)
- Tugas (T)
- Ujian Tengah Semester (UTS)
- Ujian Akhir Semester (UAS)

14. Mahasiswa dinyatakan lulus praktikum dengan nilai minimal adalah C range nilai tertera seperti pada table 1.1 berikut :

Tabel 1.3 Range nilai huruf dan angka

Huruf	Angka	Range Nilai
A	4,00	85 ≤ NA ≤ 100
	3,67	80 ≤ NA < 85
B+	3,33	75 ≤ NA < 80
B	3,00	70 ≤ NA < 75
B-	2,67	65 ≤ NA < 70
C+	2,33	60 ≤ NA < 65
C	2,00	55 ≤ NA < 60
D	1,00	40 ≤ NA < 55
E	0,00	0 ≤ NA < 40

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

E. TATA TERTIB

Tata Tertib yang harus di patuhi bai\k oleh peneliti maupun praktikan pada saat menggunakan laboratorium adalah sebagai berikut :

1. Praktikum dilaksanakan tepat waktu sesuai dengan jadwal yang ditetapkan.
2. Mahasiswa yang terlambat datang atau absen harus memberikan surat/bukti yang dapat dipercaya.
3. Mahasiswa diperkenankan pindah kelompok/jam/hari praktikum dengan syarat mengkonfirmasi H-3 sebelum pelaksanakan praktikum melalui Asisten Laboratorium.
4. Mahasiswa yang tidak hadir pada saat jadwal yang telah ditentukan diperkenankan mengikuti praktikum berikutnya dengan syarat penilaian sebelumnya akan dinilai kosong, jika ingin melakukan penambahan jam harus koordinasi dengan asisten laboratorium.
5. Mahasiswa harus meminjam alat praktikum dengan cara mengisi lembaran form pinjam alat yang tersedia.
6. Praktikum dianggap selesai jika mahasiswa telah menyerahkan laporan sementara dan alat yang dipinjam dalam keadaan baik, bersih, dan rapi.
7. Kerusakan alat yang dipinjam oleh mahasiswa menjadi tanggung jawab penuh kelompok mahasiswa yang bersangkutan.
8. Selama praktikum berlangsung, mahasiswa dilarang merokok, makan, bergurau, bermain alat, menghidupkan HandPhone, atau pun keluar masuk ruangan tanpa seijin dosen pembimbing/asisten pendamping.
9. Mahasiswa yang dinyatakan tidak lulus praktikum harus mengulang dijadwal praktikum berikutnya.
10. Laporan praktikum diserahkan ke laboratorium paling lambat 2 minggu dari pelaksanaan praktikum, jika lebih dari itu maka akan dikenakan sanksi sesuai arahan dari asisten

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

laboratorium/laboran.

F. PENGGUNAAN LABORATORIUM

Penggunaan Praktikum harus sesuai aturan yang berlaku yang sudah di jelaskan diatas dan lebih penekanan dan kesimpulannya sebagai berikut :

1. Pengguna laboratorium khususnya mahasiswa praktikum, adalah mereka yang terdaftar sebagai praktikan yang dikeluarkan oleh DA.
2. Penggunaan alat-alat praktikum sesuai dengan petunjuk kerja yang ada pada modul praktikum dengan sepengetahuan asisten laboratorium.
3. Pemimjaman kelengkapan alat-alat praktikum dengan mengisi form pemimjaman yang telah disediakan sesuai dengan modul praktikum.
4. Pelanggaran atas aturan ini dikenakan sanksi tidak dapat mengikuti praktikum berikutnya.
5. Asisten harus melaporkan terjadinya pelanggaran ke laboran dan mencatat pelanggaran yang terjadi.
6. Kerusakan karena kelalaian praktikan menjadi tanggung jawab praktikan yang bersangkutan.

G. KESEHATAN DAN KESELAMATAN KERJA

Adapun keselamatan kerja yang harus di patuhi oleh seluruh mahasiswa, asisten lab ataupun dosen sebagai berikut :

1. Pada saat akan memasuki ruang laboratorium, peneliti/mahasiswa/dosen melakukan pengisian pada *barcode* yang terdapat di depan tiap pintu laboratorium atau melalui link : bit.ly/SafetyLabTerpaduSAINTEK
2. Laboran/Asisten Laboratorium menyiapkan alat dan bahan sebelum digunakan.
3. Pengguna laboratorium diwajibkan menaati semua petunjuk keselamatan kerja dan memakai alat-alat perlindungan diri yang diwajibkan.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

4. Pengguna laboratorium mengisi logbook yang terdapat pada tiap ruang laboratorium
5. Apabila terdapat pelanggaran penggunaan Alat Pelindung Diri (APD) maupun peraturan lain yang ditetapkan, laboran/asisten laboratorium berhak memberikan sanksi.
6. Setelah selesai menggunakan laboratorium, pengguna laboratorium harus merapikan kembali dan menjaga kebersihan laboratorium.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

BAB II
KURIKULUM

A. PENDAHULUAN MATA KULIAH

Mata kuliah Algoritma dan Struktur Data merupakan salah satu matakuliah utama yang ada di Jurusan Informatika Universitas Muhammadiyah Sidoarjo dengan kode matakuliah INF23207, 4 SKS. Kompetensi pada matakuliah ini yakni: (1) Definisi dan konsep algoritma pemrograman; (2) Notasi dan struktur dasar program (review).; (3) Konsep array / list; (4) Pointer dan linked list; (5) Queue, stack, tree, graph; (6) Algoritma Sorting; (7) Algoritma Searching; (8) Penerapan berbagai macam struktur data ke dalam algoritma sorting dan searching. Tujuan dari matakuliah ini yaitu Memiliki pengetahuan yang memadai terkait cara kerja sistem komputer dan mampu menerapkan/menggunakan berbagai algoritma/metode untuk memecahkan masalah pada suatu organisasi, Kemampuan mengimplementasi kebutuhan computing dengan mempertimbangkan berbagai metode/algoritma yang sesuai..

B. TUJUAN PEMBELAJARAN

Tujuan pembelajaran mata kuliah praktikum adalah mampu menjelaskan bagaimana sistem komputer bekerja, serta bagaimana algoritma dan metode dapat diterapkan untuk memecahkan masalah pada organisasi, mampu mengimplementasikan kebutuhan computing dengan metode/algoritma yang sesuai dalam konteks praktek computing

C. PRASYARAT MATAKULIAH

Syarat teknis untuk melaksanakan praktikum Algoritma dan Struktur Data adalah Mahasiswa telah menempuh mata kuliah Algoritma dan Struktur Data (INF23207)

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

D. SILABUS PERKULIAHAN

Silabus modul praktikum mengacu pada kurikulum program studi Informatika Fakultas Sains dan Teknologi Universitas Muhammadiyah Sidoarjo pada tahun 2024 seperti berikut ini :

1. Array, Pointer, dan Struktur.
2. Linked List.
3. Stack.
4. Rekursif.
5. Sorting.

E. DIAGRAM PENCAPAIAN KOMPETENSI

Modul praktikum "Algoritma dan Struktur Data" merupakan modul yang terdapat dalam matakuliah Algoritma dan Struktur Data dengan kode INF23207, sehingga sebagai berikut ini :

Keterangan :

INF23207: Algoritma dan Struktur Data

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

F. RENCANA PELAKSANAAN PEMBELAJARAN

Skenario pembelajaran yang digunakan saat Algoritma dan Struktur Data sebagai berikut

RENCANA PELAKSANAAN PEMBELAJARAN (RPP)

TAHUN AJARAN 20_/20___

Satuan Pendidikan	**: Universitas Muhammadiyah Sidoarjo**
Mata Kuliah	**: Algoritma Dan Strukturdata**
Program studi	**: S1 Informatika**
Semester	**: 2**
Materi Pokok	**: Praktikum Algoritma Dan Strukturdata**
Alokasi waktu	**: 180 Menit**

1. Kompetensi Dasar

Mata kuliah ini memberikan pemahaman struktur data dan penerapannya menggunakan berbagai algoritma pengurutan dan pencarian data..

2. Tujuan Pembelajaran

- Mahasiswa dapat memahami Pengenalan Algoritma
- Mahasiswa dapat memahami Struktur Data Dasar
- Mahasiswa dapat memahami Penerapan struktur Data
- Mahasiswa dapat memahami Pencarian dan Pengurutan
- Mahasiswa dapat memahami Rekursi
- Mahasiswa dapat memahami Aplikasi Algoritma dan Struktur

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 - 2026

Lembar Asistensi

Praktikum Algoritma dan Struktur Data
Pokok Bahasan 1

Judul	: Struktur Data, Array, Pointer, dan Struktur
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 8
Dilaksanakan	: 05 April 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
13/5 26	13/5
(Yunianita Rahmawati, S.Kom. M.Kom)	**(Nadya Mulya Islami)**

BAB III

POKOK BAHASAN 1

Struktur Data, Array, Pointer, Dan Struktur

A. PENDAHULUAN

Pada pokok bahasan ini berisi penjelasan disertai contoh mengenai konsep struktur data, array, pointer, dan struktur yang menjadi pemahaman dasar bagi mahasiswa sebelum mempelajari struktur data, dimana konsep array, pointer, dan struktur digunakan untuk merepresentasikan sebuah struktur data

B. TUJUAN

Setelah mempelajari bab ini diharapkan mahasiswa dapat :

1. Mengetahui konsep dasar struktur data.
2. Memahami konsep array, pointer, dan struktur.

C. PENYAJIAN (TUTORIAL)

A. Konsep Dasar Struktur Data

Struktur data adalah sebuah bagian dari ilmu pemrograman dasar yang mempunyai karakteristik yang terkait dengan sifat dan cara penyimpanan sekaligus penggunaan atau pengaksesan data.

Struktur data bertujuan agar cara merepresentasikan data dalam membuat program dapat dilakukan secara efisien dalam pengolahan di memori dan pengolahan penyimpanan dari program ke storage juga lebih mudah dilakukan.

B. Konsep Dasar Array

Array adalah kumpulan elemen-elemen data. Kumpulan elemen tersebut mempunyai susunan tertentu yang teratur. Jumlah elemen terbatas, dan semua elemen mempunyai tipe data yang sama. Jenis-jenis array :

Array Satu Dimensi Struktur array satu dimensi dapat dideklarasikan dengan bentuk umum berupa : tipe var nama var [ukuran];

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Dengan :

- Tipe_var : untuk menyatakan jenis elemen array (misalnya int, char, unsigned).
- Nama_var : untuk menyatakan nama variabel yang dipakai.
- Ukuran : untuk menyatakan jumlah

maksimal elemen array. Contoh : float

nilai_ujian [5];

Array Dua Dimensi

Tipe data array dua dimensi biasa digunakan untuk menyimpan, mengolah maupun menampilkan suatu data dalam bentuk tabel atau matriks. Untuk mendeklarasikan array agar dapat menyimpan data adalah :

tipe_var nama_var [ukuran1][ukuran2];

Dimana :

- Ukuran1 menunjukkan jumlah/nomor baris.
- Ukuran2 menunjukkan jumlah/nomor kolom.

Jumlah elemen yang dimiliki array dua dimensi dapat ditentukan dari hasil perkalian : ukuran1 x ukuran 2.

Seperti halnya pada array satu dimensi, data array dua dimensi akan ditempatkan pada memori secara berurutan.

Array Multidimensi / Dimensi Banyak

Array berdimensi banyakatau multidimensi terdiri dari array yang tidak terbatas hanya dua dimensi saja. Bentuk umum pendeklarasian array multidimensi adalah :

tipe_var nama_var [ukuran1][ukuran2]...[ukuran n];

Contoh : int

data_angka

[3][6][6]; Yang

merupakan array

tiga dimensi

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Mengakses Elemen Array :

Dalam bahasa C++, data array akan disimpan dalam memori pada alokasi yang berurutan. Elemen pertama biasanya mempunyai indeks bernilai 0.

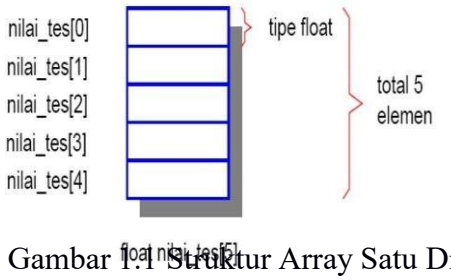
Contoh :

Float nilai_tes[5];

Jika pada contoh di atas, variabel nilai_tes mempunyai 5 elemen, maka elemen pertama mempunyai indeks sama dengan 0, elemen kedua mempunyai indeks 1, dan seterusnya. Bentuk umum pengaksesan suatu elemen variable array adalah :

Nama_var[indeks];

Gambar berikut memperlihatkan urutan komponen array dalam memori. Untuk variable array nilai_tes :



Gambar 1.1 Struktur Array Satu Dimensi

Inisialisasi Array :

Array dapat diinisialisasikan secara langsung saat pertama kali dideklarasikan (efisien untuk array berdimensi sedikit).

Contoh : int x[2]={1,2};

Array dapat dideklarasikan terlebih dahulu, baru kemudian diisi elemennya. Contoh : int x[2];

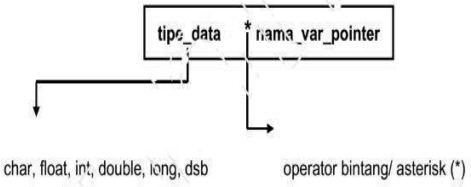
x[0]=1;

x[1]=2;

C. Konsep Dasar Pointer

Pointer adalah sebuah variabel yang berisi lamat variable yang lain. Suatu pointer dimaksudkan untuk menunjuk ke suatu alamat memori sehingga alamat dari suatu variabel dapat diketahui dengan mudah.

Deklarasi ponter :



Operator pointer :

Operator '**&**' : untuk mendapatkan **alamat memori** operand / variabel pointer. Operator '*****' : untuk mengakses **nilai data** operand / variabel pointer.

D. Konsep Dasar Struktur

Struktur adalah koleksi dari variabel yang dinyatakan dengan sebuah nama, dengan sifat setiap variabel dapat memiliki tipe yang berlainan. Struktur biasa dipakai untuk mengelompokkan beberapa informasi yang berkaitan menjadi sebuah satu kesatuan. Contoh sebuah struktur adalah informasi data tanggal, yang berisi tanggal, bulan, dan tahun.

Mendeklarasikan Struktur :

Contoh pendefinisian tipe data struktur adalah :

```
struct data_tanggal
{int tanggal;
```

Masing-masing tipe dari elemen struktur dapat berlainan. Adapun variabel_struktur1 sampai dengan variabel_struktur M menyatakan bahwa variabel struktur yang dideklarasikan bisa lebih dari satu.jika ada lebih dari satu veriabel, antara variabel struktur dipisahkan dengan tanda koma.

Mengakses Elemen Struktur :

Elemen dari struktur dapat diakses dengan menggunakan bentuk :

variabel_struktur.nama_field

Antara variabel_struktur dan nama_field dipisahkan dengan operator titik (disebut operator anggota struktur). Contoh berikut merupakan instruksi untuk mengisikan data pada field tanggal :

```
tgl_
lahi
r. ta
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
ngga
l=30
int
bul
an;
int tahun;
};
```

Yang mendefinisikan tipe struktur bernama data_tanggal, yang terdiri dari tiga buah elemen berupa tanggal, bulan, dan tahun. Bentuk umumdalam mendefinisikan dan mendeklarasikan struktur adalah :

Struct nama_tipe_struktur

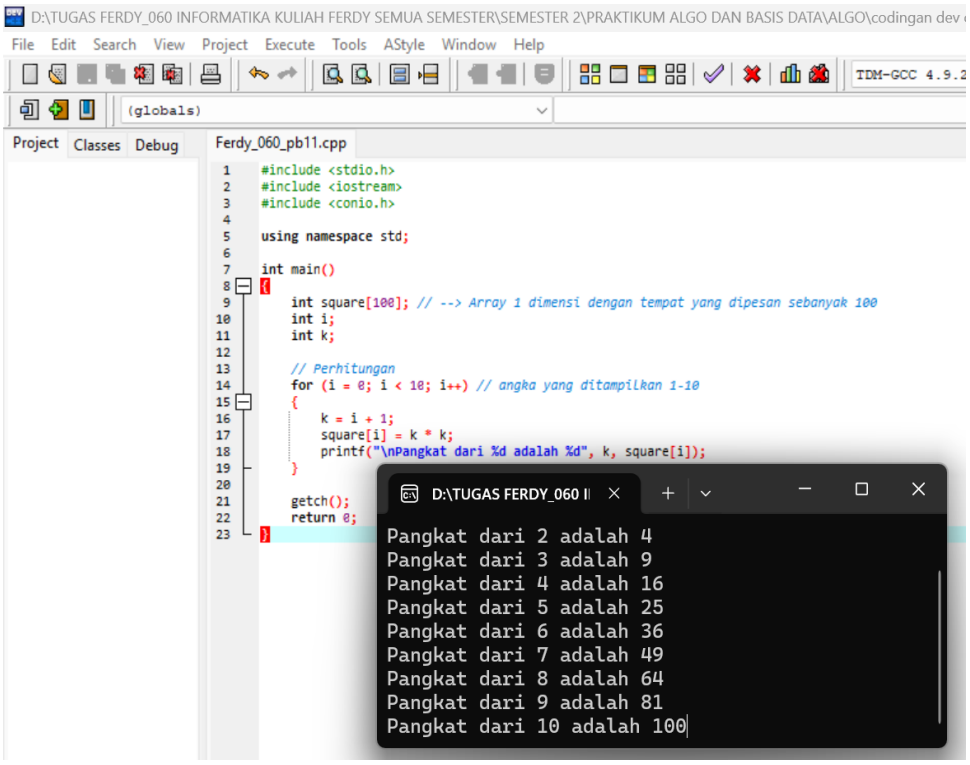
```
{
Tipe field1;
Tipe field2;
Tipe field3;
;
   Tipe field3;
}variabel_struktur1…. variabel_strukturM;
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

1. Program pangkat dengan array dimensi satu.

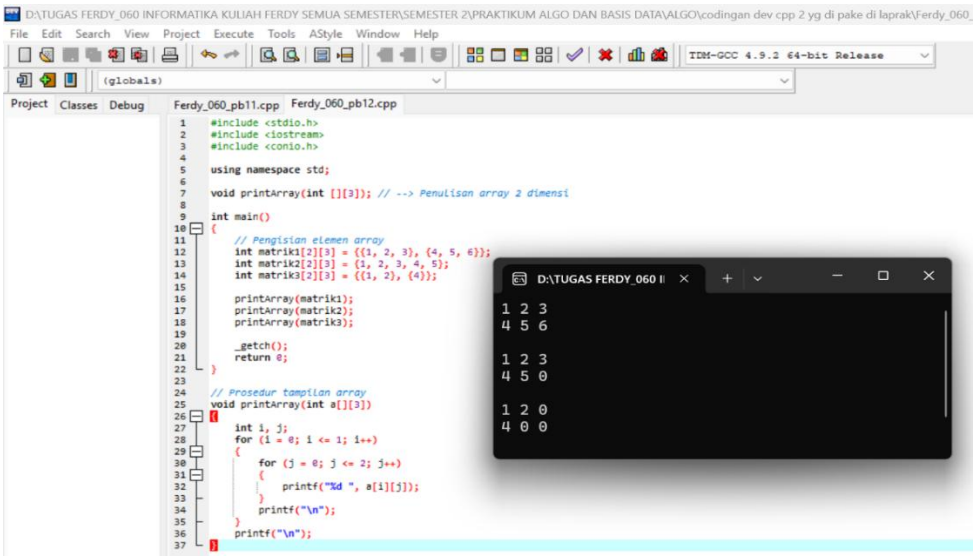
```
#include <stdio.h>
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    int square[100]; // --> Array 1 dimensi dengan tempat
yang dipesan sebanyak 100
    int i;
    int k;

    //Perhitungan
    for (i = 0; i < 10; i++) // angka yang ditampilkan 1-
10
    {
        k = i + 1;
        square[i] = k * k;
           printf("\n Pangkat dari %d adalah %d", k,
square[i]);
    }

    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

2. Program array dimensi dua

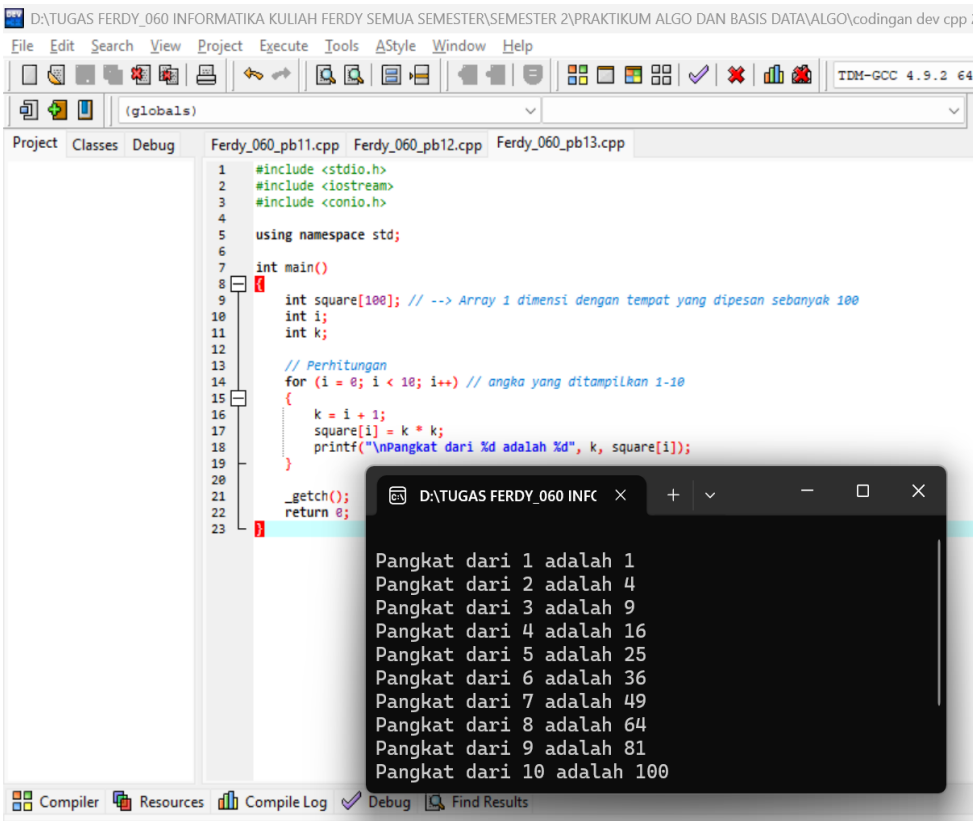
```
#include <stdio.h>
#include <iostream>
#include <conio.h>

using namespace std;

void printArray(int [][3]); // --> Penulisan array 2 dimensi

int main()
{
    // Pengisian elemen array
    int matrik1[2][3] = {{1, 2, 3}, {4, 5, 6}};
    int matrik2[2][3] = {1, 2, 3, 4, 5};
    int matrik3[2][3] = {{1, 2}, {4}};

    printArray(matrik1);
    printArray(matrik2);
    printArray(matrik3);

    _getch();
    return 0;
}

// Prosedur tampilan array
void printArray(int a[][3])
{
    int i, j;
    for (i = 0; i <= 1; i++)
    {
        for (j = 0; j <= 2; j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3. Program pointer.

```
#include <stdio.h>
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    int square[100]; // --> Array 1 dimensi dengan tempat
yang dipesan sebanyak 100
    int i;
    int k;

    // Perhitungan
    for (i = 0; i < 10; i++) // angka yang ditampilkan 1-
10
    {
        k = i + 1;
        square[i] = k * k;
        printf("\nPangkat   dari   %d   adalah   %d",   k,
square[i]);
    }

    _getch();
    return 0;
}
```

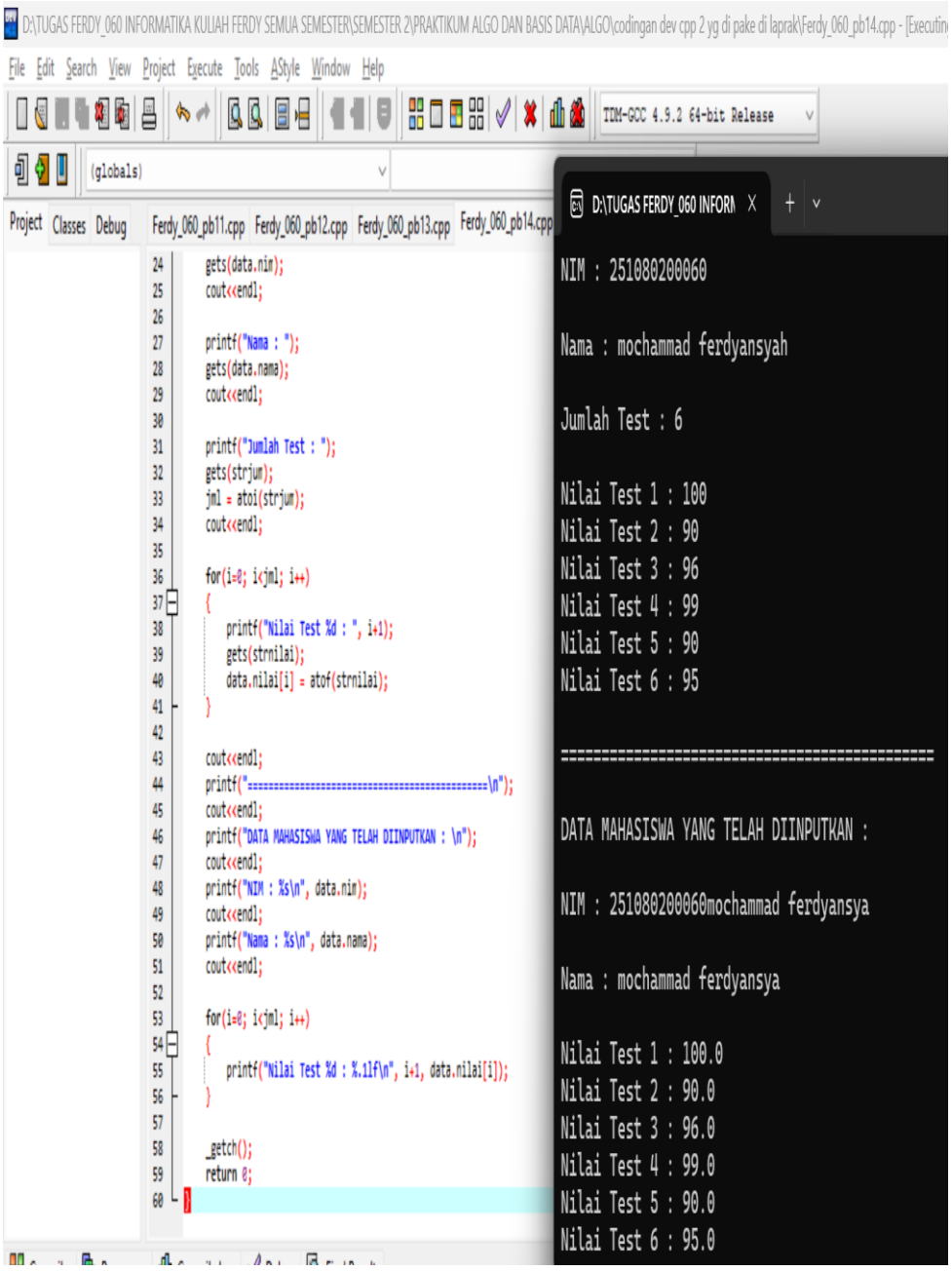


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

4. Mendeklarasikan struktur, memasukkan, dan menampilkan data struktur.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include <iostream>
#include <conio.h>

using namespace std;
#define MAX 10 //-->mx untuk nilai

struct dtnilai //-->prosedur structur
{
    char nim[12];
    char nama[20];
    double nilai[MAX];
};

int main()
{
    struct dtnilai data;
    int i, jml;
    char strnilai[5], strjum[5];

    printf("NIM : ");
    gets(data.nim);
    cout<<endl;

    printf("Nama : ");
    gets(data.nama);
    cout<<endl;

    printf("Jumlah Test : ");
    gets(strjum);
    jml = atoi(strjum);
    cout<<endl;

    for(i=0; i<jml; i++)
    {
        printf("Nilai Test %d : ", i+1);
        gets(strnilai);
        data.nilai[i] = atof(strnilai);
    }

    cout<<endl;

printf("=============================================\n")
;
    cout<<endl;
    printf("DATA MAHASISWA YANG TELAH DIINPUTKAN : \n");
    cout<<endl;
    printf("NIM : %s\n", data.nim);
    cout<<endl;
    printf("Nama : %s\n", data.nama);
    cout<<endl;

    for(i=0; i<jml; i++)
    {
        printf("Nilai   Test   %d   :   %.1lf\n",   i+1,
data.nilai[i]);
    }

    _getch();
    return 0;
}
```

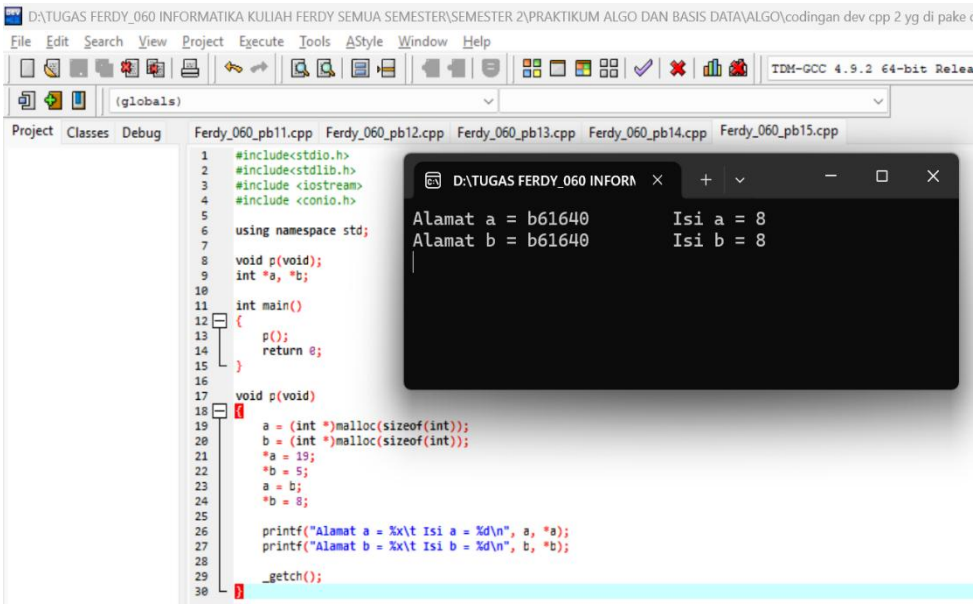
```
    for(i=0; i<jml; i++)
    {
        printf("Nilai Test %d : %.1lf\n", i+1,
data.nilai[i]);
    }

    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

5. Memesan tempat memori, mendeklarasikan, member nilai, mengcopy alamat pointer.

```
#include<stdio.h>
#include<stdlib.h>
#include <iostream>
#include <conio.h>

using namespace std;

void p(void);
int *a, *b;

int main()
{
    p();
    return 0;
}

void p(void)
{
    a = (int *)malloc(sizeof(int));
    b = (int *)malloc(sizeof(int));
    *a = 19;
    *b = 5;
    a = b;
    *b = 8;

    printf("Alamat a = %x\t Isi a = %d\n", a, *a);
    printf("Alamat b = %x\t Isi b = %d\n", b, *b);

    _getch();
}
```



Tugas!

1. Buatlah program menggunakan POINTER untuk merubah karakter yang dimasukkan dari huruf kecil menjadi huruf besar.
2. Buatlah program untuk perhitungan karakter dengan ARRAY.
3. Buatlah program array dalam struktur, yang menampilkan judul film dan tahun terbit dari data yang telah diinputkan. Data yang dimasukkan sebanyak 3 (tiga) data.

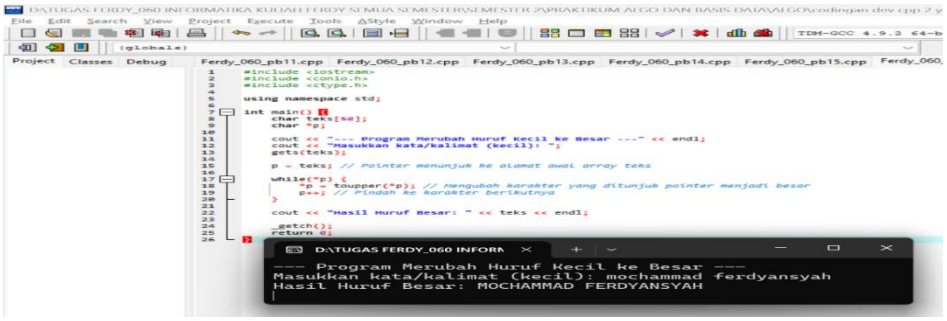
TULISKAN DAN JELASKAN SCRIPTNYA SERTA PRINTSCREEN

Jawaban :

1. Buatlah program menggunakan POINTER untuk merubah karakter yang dimasukkan dari huruf kecil menjadi huruf besar.

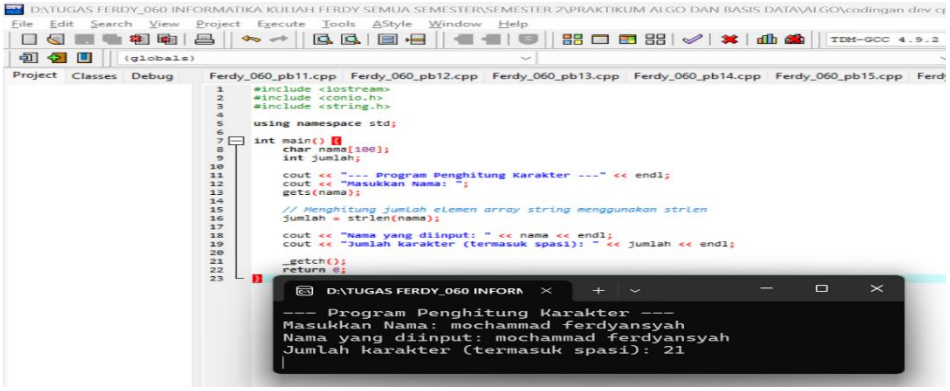
```
#include <iostream>
#include <conio.h>
#include <ctype.h>

using namespace std;

int main() {
    char teks[50];
    char *p;

    cout << "--- Program Merubah Huruf Kecil ke Besar ---" << endl;
    cout << "Masukkan kata/kalimat (kecil): ";
    gets(teks);

    p = teks; // Pointer menunjuk ke alamat awal array teks

    while(*p) {
        *p = toupper(*p); // Mengubah karakter yang ditunjuk pointer menjadi besar
        p++; // Pindah ke karakter berikutnya
    }

    cout << "Hasil Huruf Besar: " << teks << endl;

    _getch();
    return 0;
}
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271



2. Buatlah program untuk perhitungan karakter dengan ARRAY.

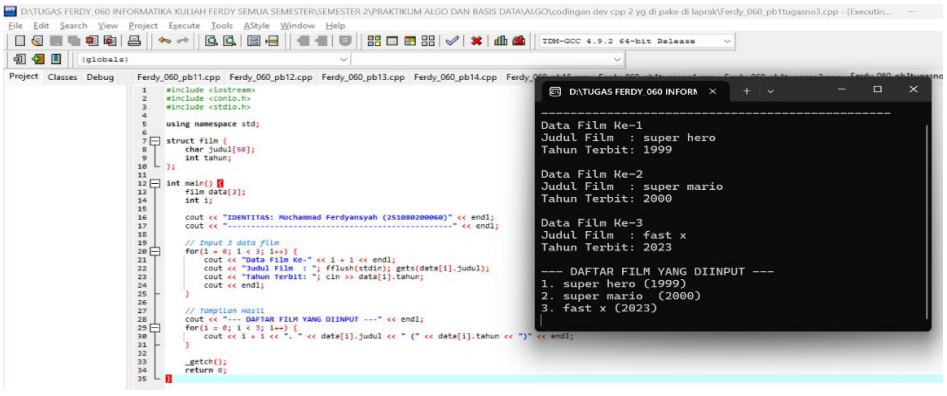
```
#include <iostream>
#include <conio.h>
#include <string.h>

using namespace std;

int main() {
    char nama[100];
    int jumlah;

    cout << "--- Program Penghitung Karakter ---" << endl;
    cout << "Masukkan Nama: ";
    gets(nama);

    // Menghitung jumlah elemen array string menggunakan
strlen
    jumlah = strlen(nama);

    cout << "Nama yang diinput: " << nama << endl;
    cout << "Jumlah karakter (termasuk spasi): " << jumlah
<< endl;

    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3. Buatlah program array dalam struktur, yang menampilkan judul film dan tahun terbit dari data yang telah diinputkan. Data yang dimasukkan sebanyak 3 (tiga) data.

```
#include <iostream>
#include <conio.h>
#include <stdio.h>

using namespace std;

struct film {
    char judul[50];
    int tahun;
};

int main() {
    film data[3];
    int i;

    cout    <<    "IDENTITAS:    Mochammad    Ferdyansyah
(251080200060)" << endl;
    cout << "-----------------------------------------
-----" << endl;

    // Input 3 data film
    for(i = 0; i < 3; i++) {
        cout << "Data Film Ke-" << i + 1 << endl;
        cout   <<   "Judul   Film     :   ";   fflush(stdin);
gets(data[i].judul);
        cout << "Tahun Terbit: "; cin >> data[i].tahun;
        cout << endl;
    }

    // Tampilan Hasil
    cout << "--- DAFTAR FILM YANG DIINPUT ---" << endl;
    for(i = 0; i < 3; i++) {
        cout << i + 1 << ". " << data[i].judul << " (" <<
data[i].tahun << ")" << endl;
    }

    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 - 2026

Lembar Asistensi

Praktikum Algoritma dan Struktur Data
Pokok Bahasan 2

Judul	: Linked List (Senarai)
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 8
Dilaksanakan	: 05 April 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
13/5 26	13/5
(Yunianita Rahmawati, S.Kom. M.Kom)	**(Nadya Mulya Islami)**

POKOK BAHASAN 2

Linked List (Senarai)

A. PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai struktur data senarai (list) yang pembahasannya meliputi definisi dan representasi list, jenis-jenis list serta operasi-operasi dasar pada list

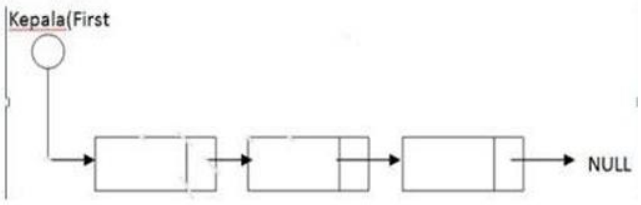
B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Menjelaskan definisi dan representasi list.
2. Mengetahui jenis-jenis list.
3. Memahami operasi-operasi pada list.

C. PENYAJIAN (TUTORIAL)

Linked List adalah sejumlah objek atau elemen yang dihubungkan satu dengan lainnya sehingga membentuk suatu list. Sedangkan objek atau elemen itu sendiri adalah merupakan gabungan beberapa data (variabel) yang dijadikan satu kelompok atau *structure* atau *record* yang dibentuk dengan perintah *struct*. Untuk menggabungkan objek satu dengan lainnya, diperlukan paling tidak sebuah variabel yang bertipe pointer. Syarat *linked list* adalah harus dapat diketahu alamat simpul pertama atau biasa dipakai variabel *First/Start/Header*. Struktur dasar sebuah list seperti gambar berikut :



Gambar 2.1 List Tunggal

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Istilah-istilah dalam *linked list* :

- Simpul

  Simpul terdiri dari dua bagian yaitu :

  a. Bagian data

  b. Bagian pointer yang menunjuk ke simpul berikutnya

- First/Header

  Variabel First/Header berisi alamat (*pointer*)/acuan (*reference*) yang menunjuk lokasi simpul pertama *linked list*, digunakan sebagai awal penelusuran *linked list*.

- Nil/Null

  Tidak bernilai, digunakan untuk menyatakan tidak mengacu ke manapun.

- Simpul Terakhir (Last)

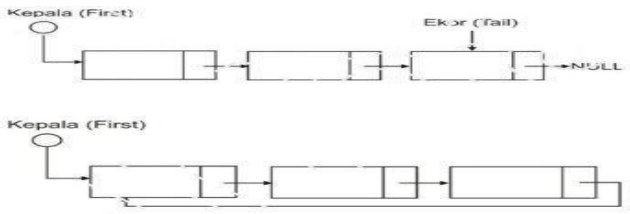
  Simpul terakhir linked list berarti tidak menunjuk simpul berikutnya. Tidak terdapat alamat disimpan di *field pointer* (bagian kedua dari simpul). Nilai null atau nil disimpan di *field pointer* di simpul terakhir.

Jenis-jenis *linked list* :

List Kosong

List kosong hanya terdiri dari sebuah penunjuk elemen yang berisi NULL (kosong), tidak memiliki satu buah elemen pun sehingga hanya berupa penunjuk awal elemen berisi NULL. List Tunggal

List tunggal adalah list yang elemennya hanya menyimpan informasi elemen setelahnya (*next*), sehingga jalannya pengaksesan list hanya dapat dilakukan secara maju. List tunggal terbagi menjadi tiga jenis yaitu list tunggal dengan kepala (*first*), list tunggal dengan kepala (*first*) dan ekor (*tail*), serta list tunggal yang berputar.

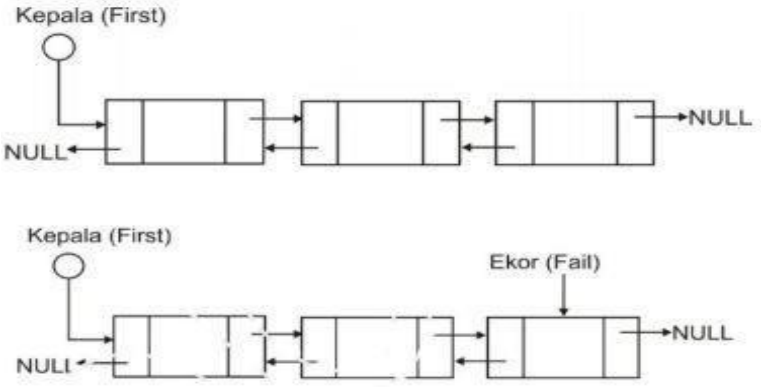


Gambar 2.2 List Tunggal dengan Kepala dan Ekor, List Tunggal Berputar

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

List Ganda

List ganda adalah sebuah list yang elemennya menyimpan informasi elemen sebelumnya dan informasi elemen setelahnya, sehingga proses penelusuran list dapat dilakukan secara maju dan mundur. List ganda terbagi menjadi tiga jenis yaitu list ganda dengan kepala (*first*), list ganda dengan kepala (*first*) dan ekor (*tail*), serta list ganda yang berputar.



Gambar 2.3 List ganda dengan Kepala, List ganda dengan Kepala dan Ekor

Operasi Dasar pada Linked List :

IsEmpty : Fungsi ini menentukan apakah linked list kosong atau tidak.

Size : operasi untuk mengirim jumlah elemen di *linked list*. *Create* : operasi untuk penciptaan list baru yang kosong.

Insertfirst : operasi untuk penyisipan simpul sebagai simpul pertama. *Insertafter* : operasi untuk penyisipan simpul setelah simpul tertentu. *Insertlast* : operasi untuk penyisipan simpul sebagai simpul terakhir.

Insertbefore : operasi untuk penyisipan simpul sebelum simpul tertentu. *Deletefirst* : operasi penghapusan simpul pertama.

Deleteafter : operasi untuk penghapusan setelah simpul tertentu.

Deletelast operasi penghapusan simpul terakhir.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

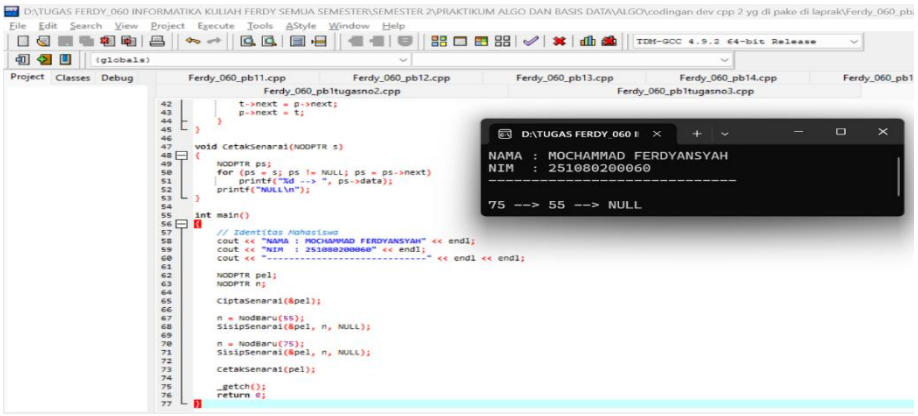
1. Contoh program sisip senarai (*linked list*).

```
#include <iostream>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>

using namespace std;

typedef struct nod
{
    int data;
    struct nod *next;
}
NOD, *NODPTR;

void CiptaSenarai(NODPTR *s)
{
    *s = NULL;
}

NODPTR NodBaru(int m)
{
    NODPTR n;
    n = (NODPTR) malloc (sizeof(NOD));
    if (n != NULL)
    {
        n->data = m;
        n->next = NULL;
    }
    return n;
}

void SisipSenarai(NODPTR *s, NODPTR t, NODPTR p)
{
    if (p == NULL)
    {
        t->next = *s;
        *s = t;
    }
    else
    {
        t->next = p->next;
        p->next = t;
    }
}

void CetakSenarai(NODPTR s)
{
    NODPTR ps;
    for (ps = s; ps != NULL; ps = ps->next)
        printf("%d --> ", ps->data);
    printf("NULL\n");
}
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
int main()
{
    // Identitas Mahasiswa
    cout << "NAMA : MOCHAMMAD FERDYANSYAH" << endl;
    cout << "NIM  : 251080200060" << endl;
    cout  <<  "-----------------------------"  <<  endl  <<
endl;

    NODPTR pel;
    NODPTR n;

    CiptaSenarai(&pel);

    n = NodBaru(55);
    SisipSenarai(&pel, n, NULL);

    n = NodBaru(75);
    SisipSenarai(&pel, n, NULL);

    CetakSenarai(pel);

    _getch();
    return 0;
}

void CetakSenarai(NODPTR s)
{
    NODPTR ps;
    for (ps = s; ps != NULL; ps = ps->next)
        printf("%d --> ", ps->data);
    printf("NULL\n");
}

int main()
{
    // Identitas Mahasiswa
    cout << "NAMA : MOCHAMMAD FERDYANSYAH" << endl;
    cout << "NIM  : 251080200060" << endl;
    cout  <<  "-----------------------------"  <<  endl  <<
endl;

    NODPTR pel;
    NODPTR n;

    CiptaSenarai(&pel);

    n = NodBaru(55);
    SisipSenarai(&pel, n, NULL);

    n = NodBaru(75);
    SisipSenarai(&pel, n, NULL);

    CetakSenarai(pel);

    _getch();
    return 0;
}
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271



2. Mendeklarasikan, memasukkan data, dan menampilkan data pada Single Linked List

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
#include <conio.h>

using namespace std;

struct dtnilai
{
    char nim[10];
    char nama[20];
    double nilai;
    struct dtnilai *next;
};

struct dtnilai *tampung;
struct dtnilai *ujung;
struct dtnilai *awal;
int j = 0;
char strnilai[5], jawab[6];

int main()
{
    // Identitas Mahasiswa
    printf("NAMA : MOCHAMMAD FERDYANSYAH\n");
    printf("NIM  : 251080200060\n");
    printf("-----------------------------\n\n");

    printf("DATA MAHASISWA : \n");

printf("=============================================\n")
;
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

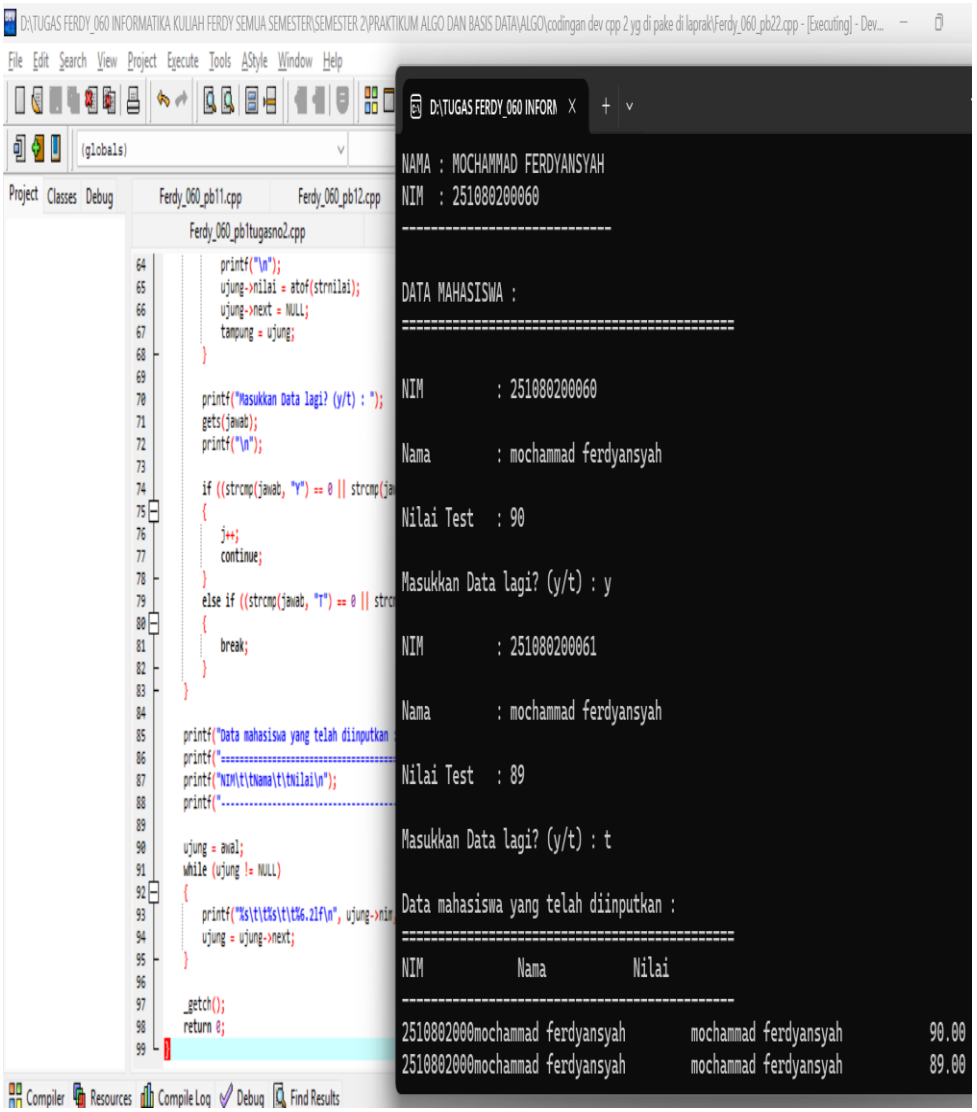
```
            printf("\n");

    while (1)
    {
        if (j == 0)
        {
            awal = (struct dtnilai*) malloc(sizeof(struct
dtnilai));
            printf("NIM          : ");
            gets(awal->nim);
            printf("\n");
            printf("Nama         : ");
            gets(awal->nama);
            printf("\n");
            printf("Nilai Test   : ");
            gets(strnilai);
            printf("\n");
            awal->nilai = atof(strnilai);
            tampung = awal;
            tampung->next = NULL;
        }
        else
        {
            ujung = (struct dtnilai*) malloc(sizeof(struct
dtnilai));
            tampung->next = ujung;
            printf("NIM          : ");
            gets(ujung->nim);
            printf("\n");
            printf("Nama         : ");
            gets(ujung->nama);
            printf("\n");
            printf("Nilai Test   : ");
            gets(strnilai);
            printf("\n");
            ujung->nilai = atof(strnilai);
            ujung->next = NULL;
            tampung = ujung;
        }

        printf("Masukkan Data lagi? (y/t) : ");
        gets(jawab);
        printf("\n");

        if ((strcmp(jawab, "Y") == 0 || strcmp(jawab, "y")
== 0))
        {
            j++;
            continue;
        }
        else if ((strcmp(jawab, "T") == 0 || strcmp(jawab,
"t") == 0))
        {
            break;
        }
    }
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
printf("Data mahasiswa yang telah diinputkan : \n");

printf("=============================================\n")
;
    printf("NIM\t\tNama\t\tNilai\n");
    printf("---------------------------------------------
\n");

    ujung = awal;
    while (ujung != NULL)
    {
        printf("%s\t\t%s\t\t%6.2lf\n", ujung->nim, ujung-
>nama, ujung->nilai);
        ujung = ujung->next;
    }

    _getch();
    return 0;
}

            tampung = awal;
            tampung->next = NULL;
        }
        else
        {
            ujung = (struct dtnilai*) malloc(sizeof(struct
dtnilai));
            tampung->next = ujung;
            printf("NIM          : ");
            gets(ujung->nim);
            printf("\n");
            printf("Nama         : ");
            gets(ujung->nama);
            printf("\n");
            printf("Nilai Test   : ");
            gets(strnilai);
            printf("\n");
            ujung->nilai = atof(strnilai);
            ujung->next = NULL;
            tampung = ujung;
        }

        printf("Masukkan Data lagi? (y/t) : ");
        gets(jawab);
        printf("\n");

        if ((strcmp(jawab, "Y") == 0 || strcmp(jawab, "y")
== 0))
        {
            j++;
            continue;
        }
        else if ((strcmp(jawab, "T") == 0 || strcmp(jawab,
"t") == 0))
        {
            break;
        }
    }
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
    printf("Data mahasiswa yang telah diinputkan : \n");

printf("=============================================\n")
;
    printf("NIM\t\tNama\t\tNilai\n");
    printf("---------------------------------------------
\n");

    ujung = awal;
    while (ujung != NULL)
    {
        printf("%s\t\t%s\t\t%6.2lf\n", ujung->nim, ujung-
>nama, ujung->nilai);
        ujung = ujung->next;
    }

    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3. Menghapus data pada simpul tertentu.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
#include <conio.h>

using namespace std;

struct dtnilai {
    char nim[15];
    char nama[20];
    double nilai;
    struct dtnilai *next;
};

int main() {
    // Identitas Mahasiswa
    printf("NAMA : MOCHAMMAD FERDYANSYAH\n");
    printf("NIM  : 251080200060\n");
    printf("----------------------------\n\n");

    struct dtnilai *tampung, *ujung, *awal, *tanda,
*tanda2;
    char strnilai[10], jawab[5], hapus[15];
    int j = 0;

    printf("DATA MAHASISWA : \n");

printf("=============================================\n")
;

    // Tahap Input Data
    while (1) {
        if (j == 0) {
            awal = (struct dtnilai*)malloc(sizeof(struct
dtnilai));
            printf("NIM          : "); gets(awal->nim);
            printf("Nama         : "); gets(awal->nama);
            printf("Nilai Test   : "); gets(strnilai);
            awal->nilai = atof(strnilai);
            tampung = awal;
            tampung->next = NULL;
        } else {
            ujung = (struct dtnilai*)malloc(sizeof(struct
dtnilai));
            tampung->next = ujung;
            printf("NIM          : "); gets(ujung->nim);
            printf("Nama         : "); gets(ujung->nama);
            printf("Nilai Test   : "); gets(strnilai);
            ujung->nilai = atof(strnilai);
            ujung->next = NULL;
            tampung = ujung;
        }
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
        printf("Masukkan Data Lagi? (y/t) : ");
        gets(jawab);
        printf("\n");

        if (strcmp(jawab, "Y") == 0 || strcmp(jawab, "y")
== 0) {
            j++;
            continue;
        } else {
            break;
        }
    }

    // Menampilkan Data Sebelum Dihapus
    printf("\nData Mahasiswa yang Diinputkan : \n");

printf("=============================================\n")
;
    printf("NIM\t\tNama\t\tNilai\n");
    ujung = awal;
    while (ujung != NULL) {
        printf("%s\t\t%s\t\t%6.2f\n",  ujung->nim,  ujung-
>nama, ujung->nilai);
        ujung = ujung->next;
    }

    // Tahap Penghapusan Data
    while (1) {
        printf("\nNIM yang akan dihapus (Ketik 't' untuk
selesai): ");
        gets(hapus);

        if (strcmp(hapus, "t") == 0 || strcmp(hapus, "T")
== 0) {
            break;
        }

        // Jika yang dihapus adalah data pertama (Head)
        if (strcmp(awal->nim, hapus) == 0) {
            tanda = awal->next;
            free(awal);
            awal = tanda;
        } else {
            // Mencari data di tengah atau akhir
            tanda2 = awal;
            tanda = tanda2->next;
            while (tanda != NULL) {
                if (strcmp(tanda->nim, hapus) == 0) {
                    if (tanda->next != NULL) {
                        tanda2->next = tanda->next;
                    } else {
                        tanda2->next = NULL;
                    }
                    free(tanda);
                    break;
                }
                tanda2 = tanda;
                tanda = tanda->next;
            }
        }
    }
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

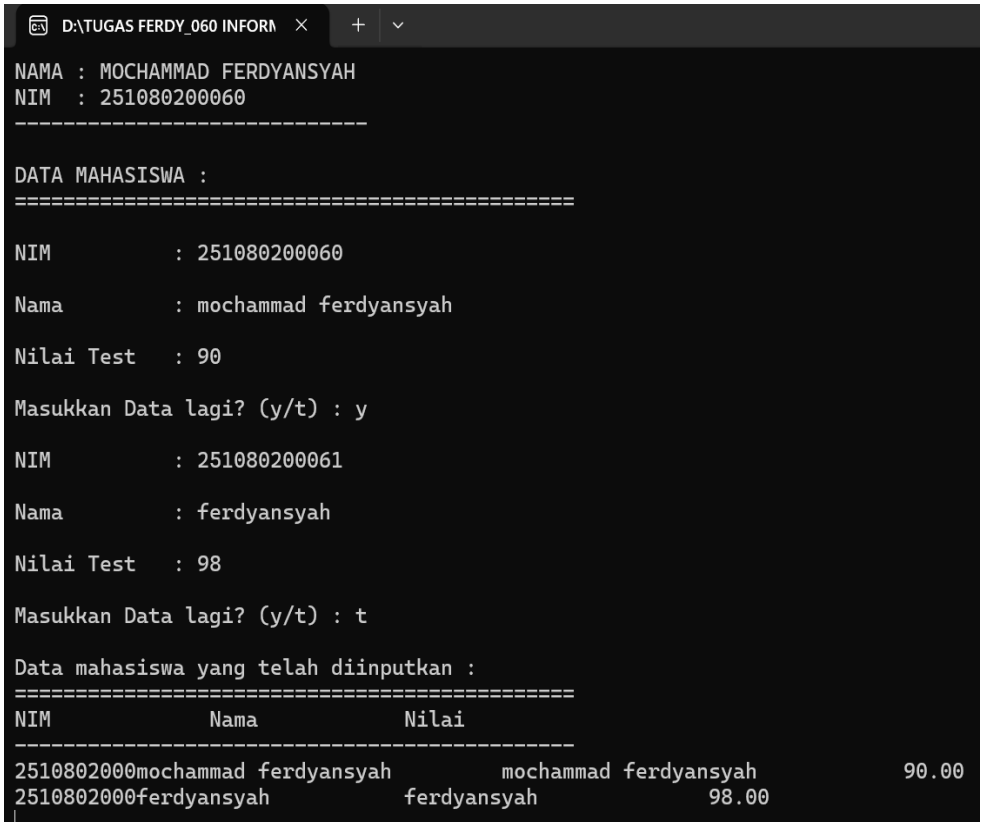
```
    // Menampilkan Data Setelah Dihapus
    printf("\nData Mahasiswa Setelah Penghapusan : \n");

printf("=============================================\n")
;
    printf("NIM\t\tNama\t\tNilai\n");
    ujung = awal;
    while (ujung != NULL) {
        printf("%s\t\t%s\t\t%6.2f\n",  ujung->nim,  ujung-
>nama, ujung->nilai);
        ujung = ujung->next;
    }

    _getch();
    return 0;
}
        printf("Masukkan Data Lagi? (y/t) : ");
        gets(jawab);
        printf("\n");

        if (strcmp(jawab, "Y") == 0 || strcmp(jawab, "y")
== 0) {
            j++;
            continue;
        } else {
            break;
        }
    }

    // Menampilkan Data Sebelum Dihapus
    printf("\nData Mahasiswa yang Diinputkan : \n");

printf("=============================================\n")
;
    printf("NIM\t\tNama\t\tNilai\n");
    ujung = awal;
    while (ujung != NULL) {
        printf("%s\t\t%s\t\t%6.2f\n",  ujung->nim,  ujung-
>nama, ujung->nilai);
        ujung = ujung->next;
    }

    // Tahap Penghapusan Data
    while (1) {
        printf("\nNIM yang akan dihapus (Ketik 't' untuk
selesai): ");
        gets(hapus);

        if (strcmp(hapus, "t") == 0 || strcmp(hapus, "T")
== 0) {
            break;
        }
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
// Jika yang dihapus adalah data pertama (Head)
        if (strcmp(awal->nim, hapus) == 0) {
            tanda = awal->next;
            free(awal);
            awal = tanda;
        } else {
            // Mencari data di tengah atau akhir
            tanda2 = awal;
            tanda = tanda2->next;
            while (tanda != NULL) {
                if (strcmp(tanda->nim, hapus) == 0) {
                    if (tanda->next != NULL) {
                        tanda2->next = tanda->next;
                    } else {
                        tanda2->next = NULL;
                    }
                    free(tanda);
                    break;
                }
                tanda2 = tanda;
                tanda = tanda->next;
            }
        }
    }

    // Menampilkan Data Setelah Dihapus
    printf("\nData Mahasiswa Setelah Penghapusan : \n");

printf("=============================================\n"
);
    printf("NIM\t\tNama\t\tNilai\n");
    ujung = awal;
    while (ujung != NULL) {
        printf("%s\t\t%s\t\t%6.2f\n",  ujung->nim,  ujung-
>nama, ujung->nilai);
        ujung = ujung->next;
    }

    _getch();
    return 0;
}
                tanda2 = tanda;
                tanda = tanda->next;
            }
        }
    }

    // Menampilkan Data Setelah Dihapus
    printf("\nData Mahasiswa Setelah Penghapusan : \n");
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
printf("=============================================\n")
;
    printf("NIM\t\tNama\t\tNilai\n");
    ujung = awal;
    while (ujung != NULL) {
        printf("%s\t\t%s\t\t%6.2f\n",   ujung->nim,   ujung-
>nama, ujung->nilai);
        ujung = ujung->next;
    }

    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tugas !

1. Buatlah program untuk menampilkan 10 bilangan secara menurun yaitu 10, 9, 8, sampai 1 dengan menggunakan LINKED LIST.
2. Buatlah sebuah program yang mengimplementasikan Linked List, dimana data yang dipakai adalah data buku yang ada dalam sebuah perpustakaan (ID Buku, Judul, dan Jumlah Buku). Program juga mengimplementasikan penambahan dan pengurangan simpul pada Linked List berdasarkan ID Buku.

TULISKAN DAN JELASKAN SCRIPTNYA SERTA PRINTSCREEN HASILNYA !!!!!

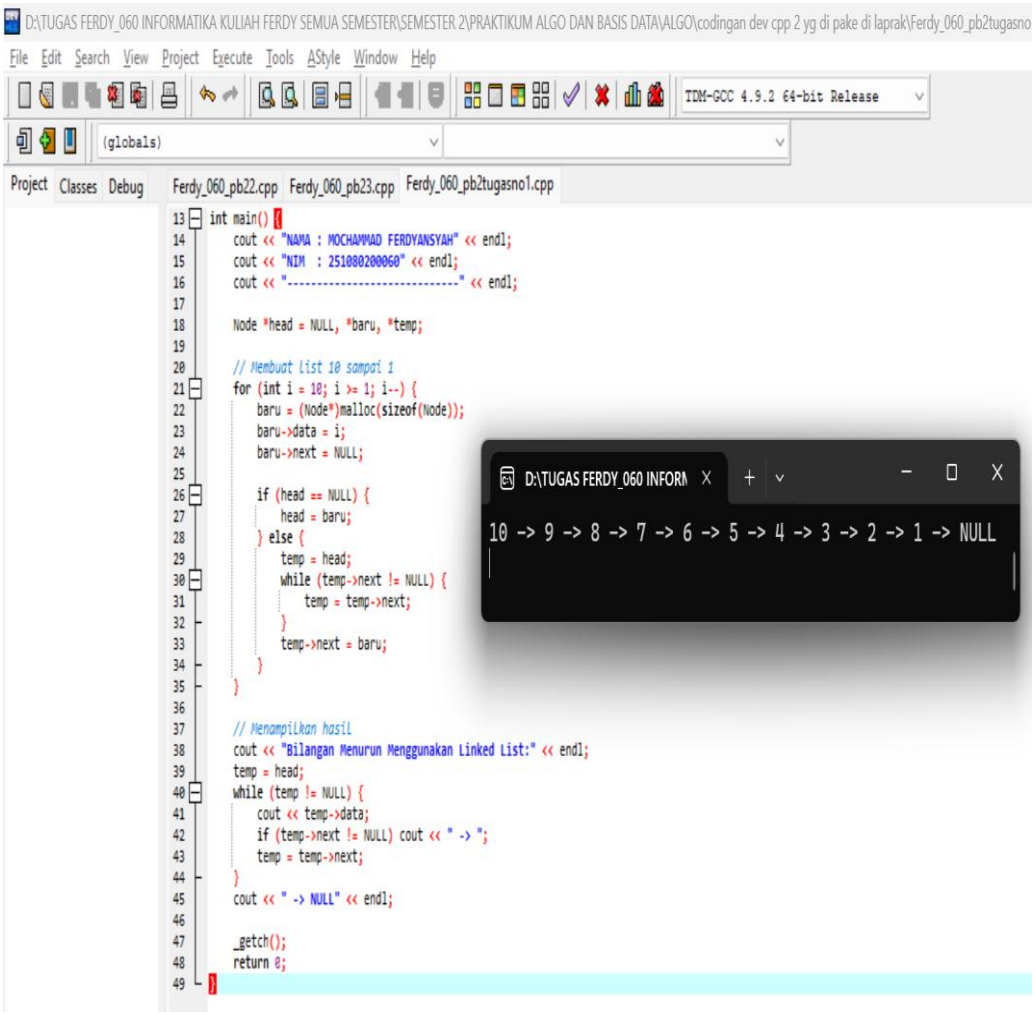
Jawaban :

1. Buatlah program untuk menampilkan 10 bilangan secara menurun yaitu 10, 9, 8, sampai 1 dengan menggunakan LINKED LIST.

```
#include <iostream>
#include <conio.h>
#include <stdlib.h>

using namespace std;

// Struktur Node
struct Node {
    int data;
    Node* next;
};

int main() {
    cout << "NAMA : MOCHAMMAD FERDYANSYAH" << endl;
    cout << "NIM  : 251080200060" << endl;
    cout << "-----------------------------" << endl;

    Node *head = NULL, *baru, *temp;

    // Membuat list 10 sampai 1
    for (int i = 10; i >= 1; i--) {
        baru = (Node*)malloc(sizeof(Node));
        baru->data = i;
        baru->next = NULL;

        if (head == NULL) {
            head = baru;
        } else {
            temp = head;
            while (temp->next != NULL) {
                temp = temp->next;
            }
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
            temp->next = baru;
        }
    }

    // Menampilkan hasil
    cout << "Bilangan Menurun Menggunakan Linked List:" <<
endl;
    temp = head;
    while (temp != NULL) {
        cout << temp->data;
        if (temp->next != NULL) cout << " -> ";
        temp = temp->next;
    }
    cout << " -> NULL" << endl;

    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

2. Buatlah sebuah program yang mengimplementasikan Linked List, dimana data yang dipakai adalah data buku yang ada dalam sebuah perpustakaan (ID Buku, Judul, dan Jumlah Buku). Program juga mengimplementasikan penambahan dan pengurangan simpul pada Linked List berdasarkan ID Buku.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
#include <conio.h>

using namespace std;

struct dtnilai {
    char id[15];
    char judul[20];
    double jumlah;
    struct dtnilai *next;
};

int main() {
    // Identitas Mahasiswa
    printf("NAMA : MOCHAMMAD FERDYANSYAH\n");
    printf("NIM  : 251080200060\n");
    printf("----------------------------\n\n");

    struct dtnilai *tampung, *ujung, *awal, *tanda, *tanda2;
    char strjumlah[10], jawab[5], hapus[15];
    int j = 0;

    printf("DATA BUKU : \n");

printf("=============================================\n");

    // Tahap Input Data
    while (1) {
        if (j == 0) {
            awal   =   (struct   dtnilai*)malloc(sizeof(struct
dtnilai));
            printf("ID           : "); gets(awal->id);
            printf("\nJudul        : "); gets(awal->judul);
            printf("\nJumlah       : "); gets(strjumlah);
            awal->jumlah = atof(strjumlah);
            tampung = awal;
            tampung->next = NULL;
        } else {
            ujung   =   (struct   dtnilai*)malloc(sizeof(struct
dtnilai));
            tampung->next = ujung;
            printf("\nID           : "); gets(ujung->id);
            printf("\nJudul        : "); gets(ujung->judul);
            printf("\nJumlah       : "); gets(strjumlah);
            ujung->jumlah = atof(strjumlah);
            ujung->next = NULL;
            tampung = ujung;
        }
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
        printf("\nMasukkan Data Lagi? (y/t) : ");
        gets(jawab);
        printf("\n");

        if (strcmp(jawab, "Y") == 0 || strcmp(jawab, "y") ==
0) {
            j++;
            continue;
        } else {
            break;
        }
    }

    // Menampilkan Data Sebelum Dihapus
    printf("\nData Buku yang telah diinputkan : \n");

printf("=============================================\n");
    printf("id\tJudul\t\tJumlah\n");
    ujung = awal;
    while (ujung != NULL) {
        printf("%s\t%s\t\t%.2f\n",  ujung->id,  ujung->judul,
ujung->jumlah);
        ujung = ujung->next;
    }

    // Tahap Penghapusan Data
    printf("\nID yang akan dihapus (ID/(t/T)) : \n");
    gets(hapus);

    if (awal != NULL && strcmp(awal->id, hapus) == 0) {
        tanda = awal->next;
        free(awal);
        awal = tanda;
    } else if (awal != NULL) {
        tanda2 = awal;
        tanda = tanda2->next;
        while (tanda != NULL) {
            if (strcmp(tanda->id, hapus) == 0) {
                tanda2->next = tanda->next;
                free(tanda);
                break;
            }
            tanda2 = tanda;
            tanda = tanda->next;
        }
    }
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
    // Menampilkan Data Setelah Dihapus
     printf("\nData Buku Yang Telah Dihapus (FIFO): \n");

printf("=============================================\n");
     printf("id\tJudul\t\tJumlah\n");
     ujung = awal;
     while (ujung != NULL) {
         printf("%s\t%s\t\t%.2f\n",   ujung->id,   ujung->judul,
ujung->jumlah);
         ujung = ujung->next;
     }

     printf("\n-----------------------------\n");
     _getch();
     return 0;
}
```

```
D:\TUGAS FERDY_060 INFORM
DATA BUKU :
=============================================
ID            : 1

Judul         : sikancil

Jumlah        : 10

Masukkan Data Lagi? (y/t) : y


ID            : 2

Judul         : anjinglaut

Jumlah        : 20

Masukkan Data Lagi? (y/t) : t


Data Buku yang telah diinputkan :
=============================================
id      Judul           Jumlah
1       sikancil                10.00
2       anjinglaut              20.00

ID yang akan dihapus (ID/(t/T)) :
2

Data Buku Yang Telah Dihapus (FIFO):
=============================================
id      Judul           Jumlah
1       sikancil                10.00
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 - 2026

Lembar Asistensi

Praktikum Algoritma dan Struktur Data
Pokok Bahasan 3

Judul	: Stack (Tumpukan)
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 05 April 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
13/5 26	13/5
(Yunianita Rahmawati, S.Kom. M.Kom)	**(Nadya Mulya Islami)**

POKOK BAHASAN 3

Stack (Tumpukan)

A. PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai mengenai struktur data tumpukan atau stack, dimana stack merupakan suatu kumpulan data yang seolah-olah ada data yang diletakkan di atas data yang lain.

B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Mahasiswa mengetahui dan memahami definisi stack.
2. Mahasiswa memahami operasi-operasi dasar stack.
3. Mahasiswa memahami representasi statis dan dinamis stack.

C. PENYAJIAN (TUTORIAL)

Stack adalah kumpulan elemen-elemen yang tersimpan dalam suatu tumpukan. Aturan penyisipan dan penghapusan elemennya tertentu:

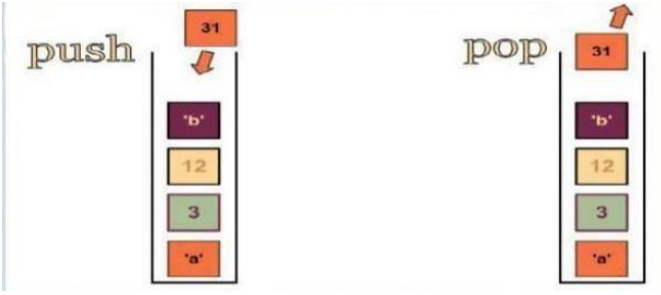
Penyisipan selalu dilakukan"di atas" TOP

Penghapusan selalu dilakukan pada TOP

Karena aturan penyisipan dan penghapusan semacam itu, TOP adalah satu-satunya alamat tempat terjadi operasi, elemen yang ditambahkan paling akhir akan menjadi elemen yang akan dihapus. Dikatakan bahwa elemen Stack tersusun secara LIFO (Last In First Out).

Seperti halnya jika kita mempunyai sebuah tumpukan buku, agar tumpukan buku itu tidak ambruk ketika kita mengambil sebuah buku di dalam tumpukan itu maka harus diambil satu per satu dari tumpukan yang paling atas dari tumpukan.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271



Gambar 3.1 Ilustrasi Stack

Perhatikan bahwa dengan definisi semacam ini, representasi tabel sangat tepat untuk mewakili stack, karena operasi penambahan dan pengurangan hanya dilakukan disalah satu ujung tabel.

Beberapa contoh penggunaan stack adalah pemanggilan prosedur, perhitungan ekspresi artimatika, rekursifitas, backtracking, penanganan interupsi, dan lain-lain.

Karakteristik penting stack sebagai berikut :

1. Elemen stack yaitu item-item data di elemen stack
2. TOP (elemen puncak dari stack)
3. Jumlah elemen pada stack
4. Status/kondisi stack, yaitu :
   - Penuh

     Bila elemen di tumpukan mencapai kapasitas maksimum tumpukan. Pada kondisi ini, tidak mungkin dilakukan penambahan ke tumpukan. Penambahan di elemen menyebabkan kondisi kesalahan Overflow.
   - Kosong

     Bila tidak ada elemen tumpukan. Pada kondisi ini, tidak mungkin dilakukan pengambilan elemen tumpukan. Pengampilan elemen menyebabkan kondisi kesalahan *Underflow*

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Stack memiliki operasi-operasi pokok sebagai berikut :

Push : Untuk menambahkan item pada tumpukan paling atas.

```
void Push(ItemType x, Stack *S)
{
    if (Full(S))
        printf("Stack FULL");
    else
    {
        S->Item[S->Count] = x;
        ++(S->Count);
    }
}
```

Pop : Untuk mengambil item teratas

```
int Pop(Stack *S, ItemType *x)
{
    if (Empty(S))
        printf("Stack Kosong");
    else
    {
        --(S->Count);
        *x = S->Item[S->Count];
    }
}
```

Clear : Untuk mengosongkan stack

```
void
InitializeS
tack  (Stack
S)
{
      S->Count=0;
}
```

IsEmpty: Untuk memeriksa apakah stack kosong int Empty (Stack*S)

```
{
      return (S->Count==0);
}
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Is Full : Untuk memeriksa apakah stack sudah penuh

```
int Full(Stack *S)
{
    return (S->Count == MAXSTACK);
}
```

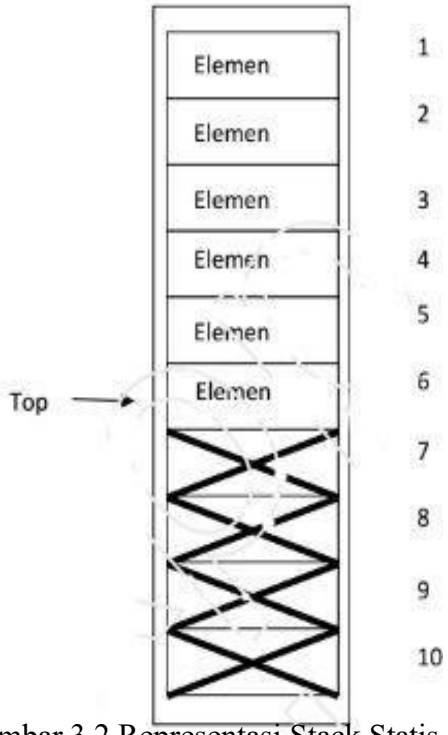
Representasi stack:

- Representasi statis

  Stack dengan representasi statis biasanya diimplementasikan dengan menggunakan array. Sebuah array memiliki tempat yang dialokasikan diawal sehingga sebuah elemen yang dimasukkan dalam sebuah array terbatas pada tempat yang ada pada array. Karena menggunakan array maka stack dengan representasi statis dalam mengalami kondisi elemen penuh. Ilustrasi stack dengan representasi statis dapat dilihat pada gambar 3.2 :

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

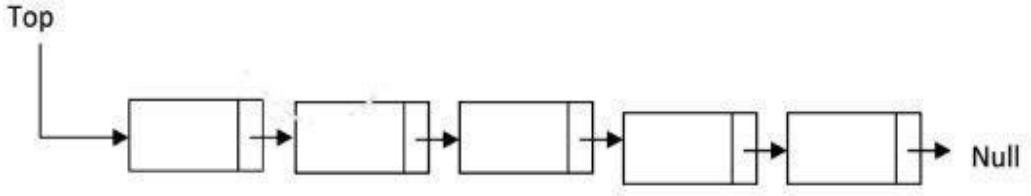
Stack dengan representasi statis biasanya diimplementasikan dengan menggunakan array. Sebuah array memiliki tempat yang dialokasikan diawal sehingga sebuah elemen yang dimasukkan dalam sebuah array terbatas pada tempat yang ada pada array. Karena menggunakan array maka stack dengan representasi statis dalam mengalami kondisi elemen penuh. Ilustrasi stack dengan representasi statis dapat dilihat pada gambar 3.2 :



Gambar 3.2 Representasi Stack Statis

Representasi dinamis

Stack dengan representasi dinamis biasanya diimplementasikan dengan menggunakan pointer yang menunjuk pada elemen-elemen yang dialokasikan pada memori. Ilustrasi stack dengan representasi dinamis dapat dilihat pada gambar 3.3 :



Gambar 3.3 Representasi Stack Dinamis

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Karena semua operasi pada sebuah stack diawali dengan elemen yang paling atas maka jika menggunakan representasi dinamis saat elemen ditambahkan akan menggunakan penambahan elemen pada awal stack (addfirst) dan saat pengambilan atau penghapusan elemen menggunakan penghapusan di awal stack (delfirst).

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

1) Program Stack :

```
#include <stdio.h>
#include <conio.h>
#include <iostream>

using namespace std;

#define MAXSTACK 3

typedef int itemType;

typedef struct
{
    int item[MAXSTACK];
    int jml;
} Stack;

void init(Stack *s)
{
    s->jml = 0;
}

int kosong(Stack *s)
{
    return (s->jml == 0);
}

int penuh(Stack *s)
{
    return (s->jml == MAXSTACK);
}

void isi(itemType x, Stack *s)
{
    if (penuh(s))
        printf("\nMAAF!!! Data PENUH\n");
    else {
        s->item[s->jml] = x;
        ++(s->jml);
    }
}

void ambil(Stack *s, itemType *x)
{
    if (kosong(s))
        printf("\nMaaf Data Kosong\n");
    else
    {
        --(s->jml);
        *x = s->item[s->jml];
        s->item[s->jml] = 0;
        printf("\nData %i Berhasil Diambil\n", *x);
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
    }
}

void tampil(Stack *s)
{
    if (kosong(s))
        printf("\nMaaf Data Masih Kosong\n");
    else
    {
        printf("\n");
        for (int i = s->jml - 1; i >= 0; i--)
        {
            printf("Data: %d\n", s->item[i]);
        }
    }
}

void hapus(Stack *s)
{
    s->jml = 0;
    printf("\nSemua Data Berhasil Dihapus\n");
}

int main()
{
    // Identitas Mahasiswa
    printf("NAMA : MOCHAMMAD FERDYANSYAH\n");
    printf("NIM  : 251080200060\n");
    printf("----------------------------\n\n");

    int pil;
    Stack tumpukan;
    itemType data;

    init(&tumpukan);

    do
    {
        printf("\nMENU: \n 1. Isi (Data Angka)\n 2. Ambil\n
3. Lihat\n 4. Hapus (Hapus Semua Data)\n 5. Keluar\n");
        printf("\nMasukkan Pilihan : ");
        scanf("%i", &pil);

        switch (pil)
        {
        case 1:
            printf("\nMasukkan Data Angka : ");
            scanf("%i", &data);
            isi(data, &tumpukan);
            break;
        case 2:
            ambil(&tumpukan, &data);
            break;
        case 3:
            tampil(&tumpukan);
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
            break;
        case 4:
            hapus(&tumpukan);
            break;
        }
    } while (pil != 5);

    _getch();
    return 0;
}
```

```
D:\TUGAS FERDY_060 INFORM

NAMA : MOCHAMMAD FERDYANSYAH
NIM  : 251080200060
----------------------------


MENU:
 1. Isi (Data Angka)
 2. Ambil
 3. Lihat
 4. Hapus (Hapus Semua Data)
 5. Keluar

Masukkan Pilihan : 1

Masukkan Data Angka : 20

MENU:
 1. Isi (Data Angka)
 2. Ambil
 3. Lihat
 4. Hapus (Hapus Semua Data)
 5. Keluar

Masukkan Pilihan : 3

Data: 20

MENU:
 1. Isi (Data Angka)
 2. Ambil
 3. Lihat
 4. Hapus (Hapus Semua Data)
 5. Keluar

Masukkan Pilihan : 2

Data 20 Berhasil Diambil

MENU:
 1. Isi (Data Angka)
 2. Ambil
 3. Lihat
 4. Hapus (Hapus Semua Data)
 5. Keluar

Masukkan Pilihan : 10

MENU:
 1. Isi (Data Angka)
 2. Ambil
 3. Lihat
 4. Hapus (Hapus Semua Data)
 5. Keluar

Masukkan Pilihan :
```

```
Masukkan Pilihan : 10

MENU:
 1. Isi (Data Angka)
 2. Ambil
 3. Lihat
 4. Hapus (Hapus Semua Data)
 5. Keluar

Masukkan Pilihan : 4

Semua Data Berhasil Dihapus

MENU:
 1. Isi (Data Angka)
 2. Ambil
 3. Lihat
 4. Hapus (Hapus Semua Data)
 5. Keluar

Masukkan Pilihan : 3

Maaf Data Masih Kosong

MENU:
 1. Isi (Data Angka)
 2. Ambil
 3. Lihat
 4. Hapus (Hapus Semua Data)
 5. Keluar

Masukkan Pilihan : 5
```

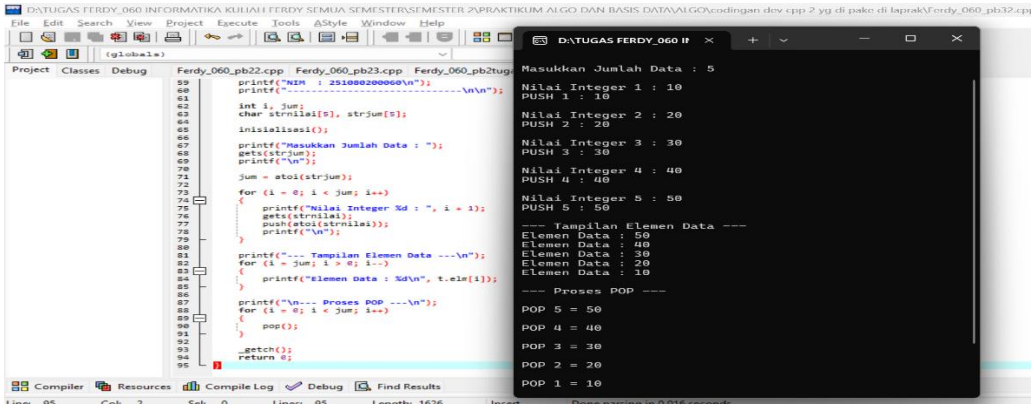

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

2) Mendeklarasi, memasukkan, dan menampilkan elemen data pada sebuah stack dengan menggunakan array :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <iostream>
#include <conio.h>

using namespace std;

#define MAX 10

struct ttumpukan
{
    int elm[MAX];
    int puncak;
};

struct ttumpukan t;
int top;

void inisialisasi()
{
    t.puncak = 0;
    top = 0;
}

void push(int x)
{
    if (t.puncak == MAX)
    {
        printf("Stack Sudah Penuh\n");
    }
    else
    {
        t.puncak = top + 1;
        t.elm[t.puncak] = x;
        top = t.puncak;
        printf("PUSH %d : %d\n", t.puncak, t.elm[t.puncak]);
    }
}

void pop()
{
    if (top == 0)
    {
        printf("Stack Kosong\n");
    }
    else
    {
        printf("\nPOP %d = %d\n", top, t.elm[top]);
        t.puncak = top - 1;
        top = t.puncak;
    }
}
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
int main()
{
    // Identitas Mahasiswa
    printf("NAMA : MOCHAMMAD FERDYANSYAH\n");
    printf("NIM  : 251080200060\n");
    printf("-----------------------------\n\n");

    int i, jum;
    char strnilai[5], strjum[5];

    inisialisasi();

    printf("Masukkan Jumlah Data : ");
    gets(strjum);
    printf("\n");

    jum = atoi(strjum);

    for (i = 0; i < jum; i++)
    {
        printf("Nilai Integer %d : ", i + 1);
        gets(strnilai);
        push(atoi(strnilai));
        printf("\n");
    }

    printf("--- Tampilan Elemen Data ---\n");
    for (i = jum; i > 0; i--)
    {
        printf("Elemen Data : %d\n", t.elm[i]);
    }

    printf("\n--- Proses POP ---\n");
    for (i = 0; i < jum; i++)
    {
        pop();
    }

    _getch();
    return 0;
}
```



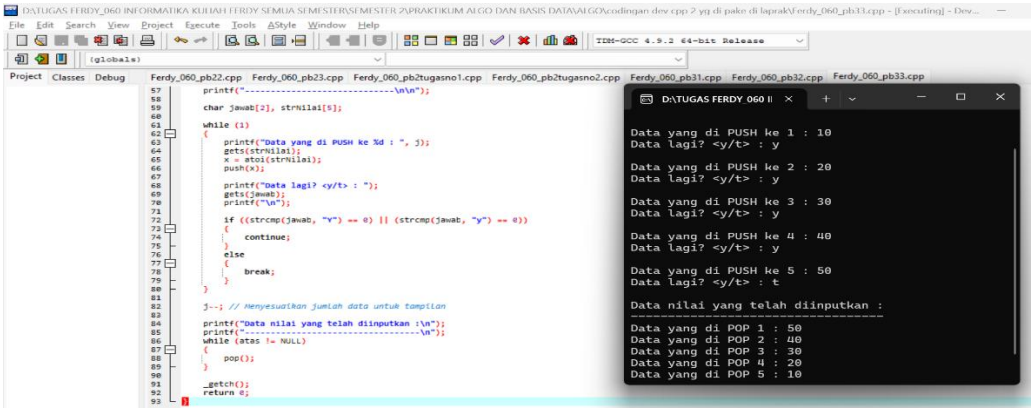
fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3) Merepresentasikan stack menggunakan Linked List :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <iostream>
#include <conio.h>

using namespace std;

struct sStack
{
    int isi;
    struct sStack *next;
};

int x, i = 1, j = 1;
struct sStack *atas, *bantu, *baru;

void push(int nilai)
{
    if (j == 1)
    {
        atas = (struct sStack*)malloc(sizeof(struct sStack));
        atas->isi = nilai;
        atas->next = NULL;
    }
    else
    {
        baru = (struct sStack*)malloc(sizeof(struct sStack));
        baru->isi = nilai;
        baru->next = atas;
        atas = baru;
    }
    j++;
}

void pop()
{
    if (atas == NULL)
    {
        printf("Stack Kosong\n");
    }
    else
    {
        bantu = atas;
        printf("Data yang di POP %d : %d\n", i, bantu->isi);
        atas = atas->next;
        i++;
        free(bantu);
    }
}

int main()
{
    // Identitas Mahasiswa
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
    // Identitas Mahasiswa
    printf("NAMA : MOCHAMMAD FERDYANSYAH\n");
    printf("NIM  : 251080200060\n");
    printf("-----------------------------\n\n");

    char jawab[2], strNilai[5];

    while (1)
    {
        printf("Data yang di PUSH ke %d : ", j);
        gets(strNilai);
        x = atoi(strNilai);
        push(x);

        printf("Data lagi? <y/t> : ");
        gets(jawab);
        printf("\n");

        if ((strcmp(jawab, "Y") == 0) || (strcmp(jawab, "y")
== 0))
        {
            continue;
        }
        else
        {
            break;
        }
    }

    j--; // Menyesuaikan jumlah data untuk tampilan

    printf("Data nilai yang telah diinputkan :\n");
    printf("---------------------------------\n");
    while (atas != NULL)
    {
        pop();
    }

    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

TUGAS

1. Buatlah program stack statis yang menampilkan data yang di–*push* dan di-*pop*.
2. Diketahui data-data
   berikut : T, I, D, dan E.
   Buatlah program untuk memasukkan data-data di atas sehingga akan muncul tampilan berikut : T, E, D, I.
   Manfaatkan operasi PUSH dan POP.

TULISKAN DAN JELASKAN SCRIPTNYA SERTA

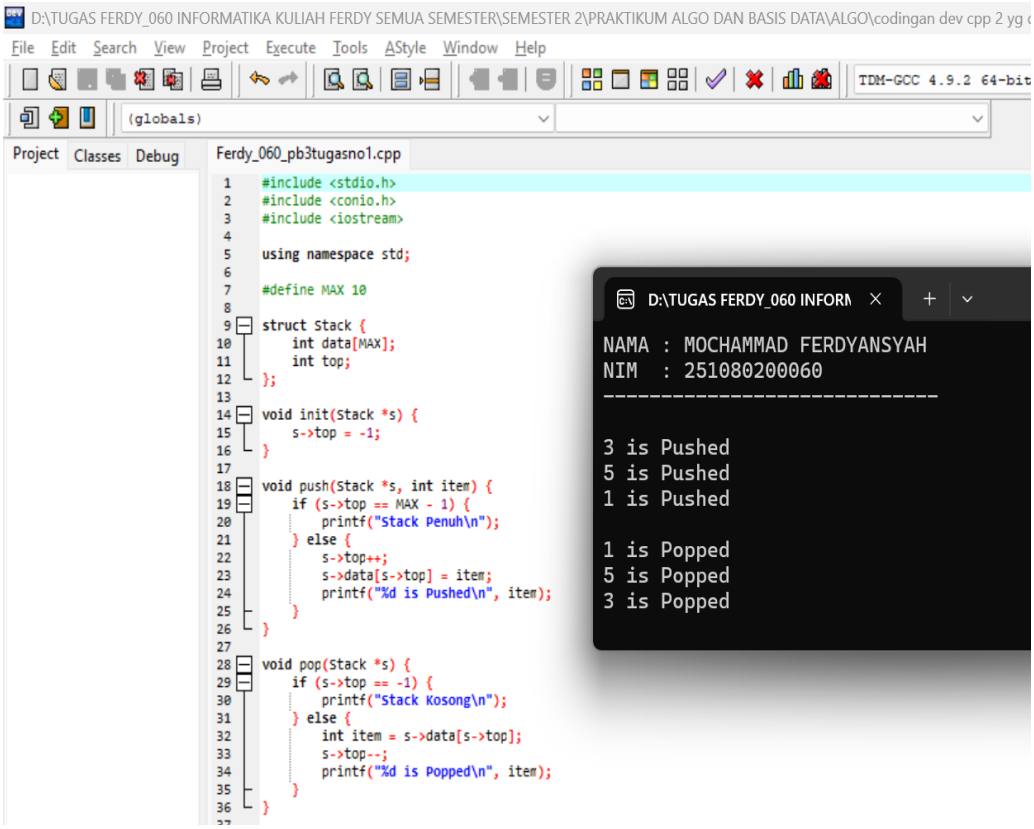
Jawaban

1. Buatlah program stack statis yang menampilkan data yang di–push dan di-pop.

```
#include <stdio.h>
#include <conio.h>
#include <iostream>

using namespace std;

#define MAX 10

struct Stack {
    int data[MAX];
    int top;
};

void init(Stack *s) {
    s->top = -1;
}

void push(Stack *s, int item) {
    if (s->top == MAX - 1) {
        printf("Stack Penuh\n");
    } else {
        s->top++;
        s->data[s->top] = item;
        printf("%d is Pushed\n", item);
    }
}

void pop(Stack *s) {
    if (s->top == -1) {
        printf("Stack Kosong\n");
    } else {
        int item = s->data[s->top];
        s->top--;
        printf("%d is Popped\n", item);
    }
}
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```c
int main() {
    printf("NAMA : MOCHAMMAD FERDYANSYAH\n");
    printf("NIM  : 251080200060\n");
    printf("----------------------------\n\n");

    Stack tumpukan;
    init(&tumpukan);

    // Proses sesuai urutan output yang kamu inginkan
    push(&tumpukan, 3);
    push(&tumpukan, 5);
    push(&tumpukan, 1);

    printf("\n");

    pop(&tumpukan);
    pop(&tumpukan);
    pop(&tumpukan);

    _getch();
    return 0;
}
```



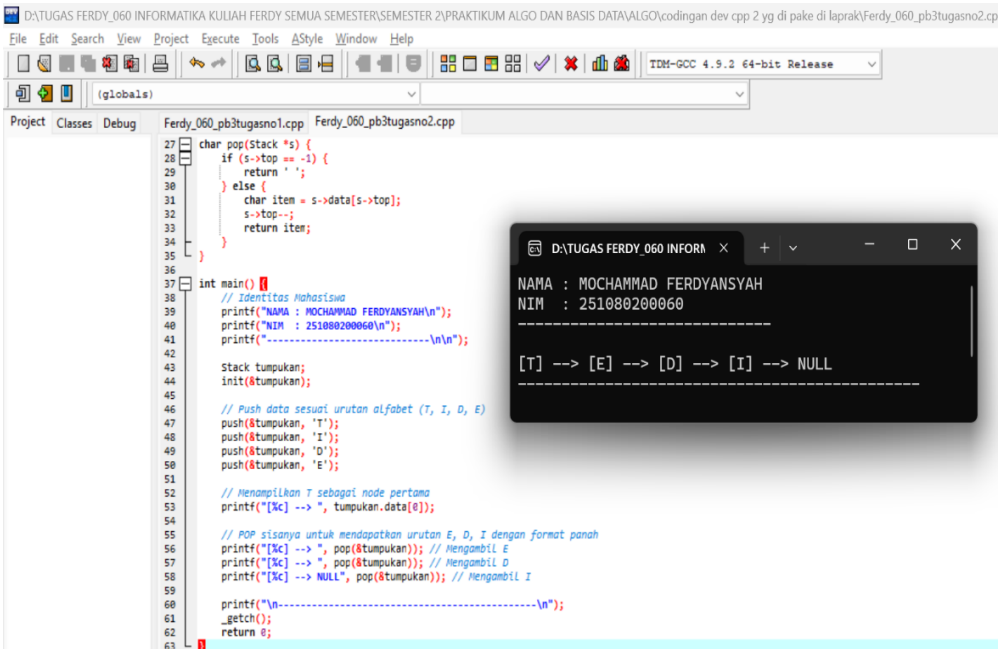
fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Diketahui data-data berikut : T, I, D, dan **x**E. Buatlah program untuk memasukkan data-data di atas sehingga akan muncul tampilan berikut : T, E, D, I. Manfaatkan operasi PUSH dan POP.

```
#include <stdio.h>
#include <conio.h>
#include <iostream>

using namespace std;

#define MAX 10

struct Stack {
    char data[MAX];
    int top;
};

void init(Stack *s) {
    s->top = -1;
}

void push(Stack *s, char item) {
    if (s->top == MAX - 1) {
        printf("Stack Penuh\n");
    } else {
        s->top++;
        s->data[s->top] = item;
    }
}

char pop(Stack *s) {
    if (s->top == -1) {
        return ' ';
    } else {
        char item = s->data[s->top];
        s->top--;
        return item;
    }
}

int main() {
    // Identitas Mahasiswa
    printf("NAMA : MOCHAMMAD FERDYANSYAH\n");
    printf("NIM  : 251080200060\n");
    printf("-----------------------------\n\n");

    Stack tumpukan;
    init(&tumpukan);

    // Push data sesuai urutan alfabet (T, I, D, E)
    push(&tumpukan, 'T');
    push(&tumpukan, 'I');
    push(&tumpukan, 'D');
    push(&tumpukan, 'E');
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
    // Menampilkan T sebagai node pertama
    printf("[%c] --> ", tumpukan.data[0]);

    // POP sisanya untuk mendapatkan urutan E, D, I dengan
format panah
    printf("[%c] --> ", pop(&tumpukan)); // Mengambil E
    printf("[%c] --> ", pop(&tumpukan)); // Mengambil D
    printf("[%c] --> NULL", pop(&tumpukan)); // Mengambil I

    printf("\n----------------------------------------------
-\n");
    _getch();
    return 0;
}
```



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 - 2026

Lembar Asistensi

Praktikum Algoritma dan Struktur Data
Pokok Bahasan 4

Judul	: Queue (Antrian)
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 30 Juni 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
(Yunianita Rahmawati, S.Kom. M.Kom)	**(Nadya Mulya Islami)**

POKOK BAHASAN 4

QUEUE (ANTRIAN)

A. PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai antrian atau queue, dimana struktur data hamper sama dengan tumpukan atau stack yang merupakan struktur data yang linier. Perbedaanya adalah operasi penambahan dan pengurangan pada ujung yang berbeda.

B. TUJUAN

Setelah mempelajari materi ini diharapkan mahasiswa mampu:

1. Mengetahui dan memahami definisi antrian.
2. Memahami operasi-operasi dasar pada antrian.
3. Memahami representasi statis dan dinamis pada antrian.

C. PENYAJIAN (TUTORIAL)

Antrian adalah suatu kumpulan data yang penambahan elemen nya hanya bisa dilakukan pada suatu ujung (disebut sisi belakang atau REAR), dan penghapusan atau pengambilan elemen dilakukan lewat ujung yang lain (disebut sisi depan atau FRONT). Prinsip yang digunakan dalam antrian ini adalah FIFO *(First In First Out)* yaitu elemen yang pertama kali masuk akan keluar pertama kalinya.

Penggunaan antrian antara lain simulasi antrian di dunia nyata (antrian pembelian tiket), sistem jaringan komputer (pemrosesan banyak paket yang dating dari banyak koneksi pada *host, bridge, gateway*), dan lain-lain.

Gambar 4.1 Ilustrasi Antrian dengan 8 Elemen

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Karakteristik penting antrian sebagai berikut :

a Elemen antrian yaitu item-item data yang terdapat dalam antrian.

b *Head/front* (elemen terdepan antrian).

c *Tail/rear* (elemen terakhir antrian).

d Jumlah antrian pada antrian (*count*).

e Status/kondisi antrian, ada dua yaitu :

- Penuh

  Bila elemen di antrian mencapai kapasitas maksimum antrian. Pada kondisi ini, tidak mungkin dilakukan penambahan ke antrian. Penambahan di elemen menyebabkan kondisi kesalahan *Overflow.*

- Kosong

  Bila tidak ada elemen antrian. Pada kondisi ini, tidak mungkin dilakukan pengambilan elemen antrian. Pengambilan elemen menyebabkan kondisi kesalahan *Underflow.*

Operasi-operasi pokok pada antrian diantaranya adalah :

1. Create Membuat antrian baru.

   NOEL(CREATE(Q))=0

   FRONT(CREATE(Q))=tidak terdefinisi

   REAR(CREATE(Q)=tidak terdefinisi

2. IsEmpty Untuk memeriksa apakah antrian sudah penuh atau belum. ISEMPTY(Q)=True, jika Q adalah queue penuh.

3. IsFull Mengecek apakah antrian sudah penuh atau belum. ISFULL(Q)=True, jika Q adalah queue penuh.

4. Enqueue/Insert Menambahkan elemen ke dalam antrian, penambahan elemen selalu ditambahkan di dalam elemen paling belakang. REAR(INSERT(A,Q))=A

   ISEMPTY(INSERT(A,Q))=FALSE

   Algoritma QINSERT :

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

a IF FRONT = 1 AND REAR = N, OR IF FRONT = REAR + 1, THEN OVERFLOW, RETURN

b IF FRONT := NULL, THEN
SET FRONT := 1 AND
REAR := 1 ELSE IF REAR =
N,
THEN SET REAR :=
1 ELSE
SET REAR := REAR + !

c SET QUEUE[REAR] := ITEM

d RETURN

5. Dequeue / Remove Untuk menghapus elemen terdepan/pertama dari antrian. Algoritma QDELETE :

a IF FRONT := NULL, THEN UNDERFLOW, RETURN

b SET ITEM := QUEUE{FRONT]

c [FIND NEW VALUE OF FRONT] IF FRONT = REAR,
THEN SET FRONT := NULL AND REAR
;=NULL ELSE IF FRONT = N,
THEN SET FRONT := 1
ELSE
SET FRONT := FRONT+!

d RETURN

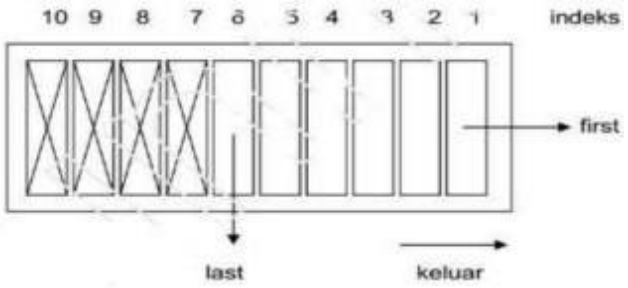
Representasi Queue :

Representasi statis

Queue dengan representasi statis biasanya diimplementasikan dengan menggunakan array. Sebuah array memiliki tempat yang dialokasikan di awal sehingga sebuah elemen yang dimasukkan dalam sebuah array terbatas pada tempat yang ada pada array. Karena menggunakan array maka queue dengan representasi statis dalam mengalami kondisi elemen penuh. Ilustrasi

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

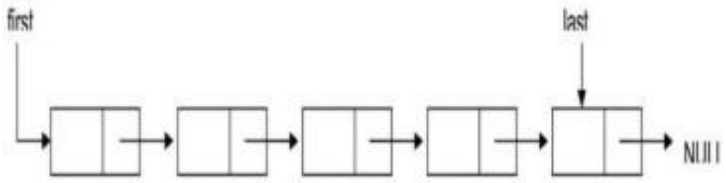
queue dengan representasi statis dapat dilihat pada gambar :



Gambar 4.2 Representasi Queue Statis

Representasi dinamis

Queue dengan representasi dinamis biasanya diimplementasikan dengan menggunakan pointer yang menunjuk pada elemen-elemen yang dialokasikan pada memori. Ilustrasi queue dengan representasi dinamis dapat dilihat pada gambar :



Gambar 4.3 Representasi Queue Dinamis

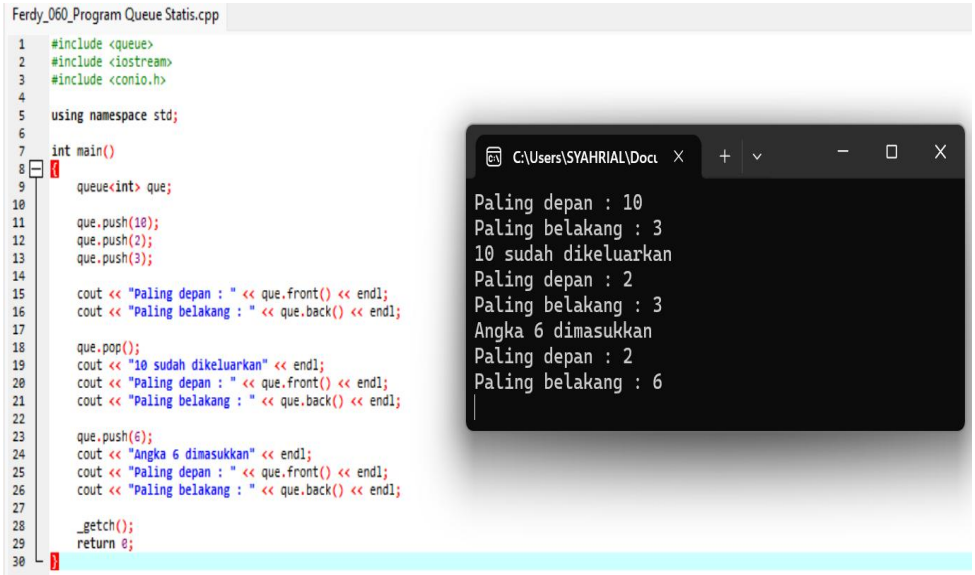
fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

1. Program Queue Statis.

```
#include <queue>
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    queue<int> que;

    que.push(10);
    que.push(2);
    que.push(3);

    cout << "Paling depan : " << que.front() << endl;
    cout << "Paling belakang : " << que.back() << endl;

    que.pop();
    cout << "10 sudah dikeluarkan" << endl;
    cout << "Paling depan : " << que.front() << endl;
    cout << "Paling belakang : " << que.back() << endl;

    que.push(6);
    cout << "Angka 6 dimasukkan" << endl;
    cout << "Paling depan : " << que.front() << endl;
    cout << "Paling belakang : " << que.back() << endl;

    _getch();
    return 0;
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

2. Program Queue.

```
#include <iostream>
#include    <cstdlib>    //    Tambahan    library    untuk
system("pause")

using namespace std;
#define MAX 5

class queue
{
private:
    int t[MAX];
    int al;
    int dl;

public:
    queue()
    {
        dl=-1;
        al=-1;
    }

    void del()
    {
        int tmp;
        if(dl==-1)
        {
            cout<<"Queue kosong";
        }
        else
        {
            for(int j=0;j<=al;j++)
            {
                if((j+1)<=al)
                {
                    tmp=t[j+1];
                    t[j]=tmp;
                }
                else
                {
                    al--;
                    if(al==-1)
                        dl=-1;
                    else
                        dl=0;
                }
            }
        }
    }
```

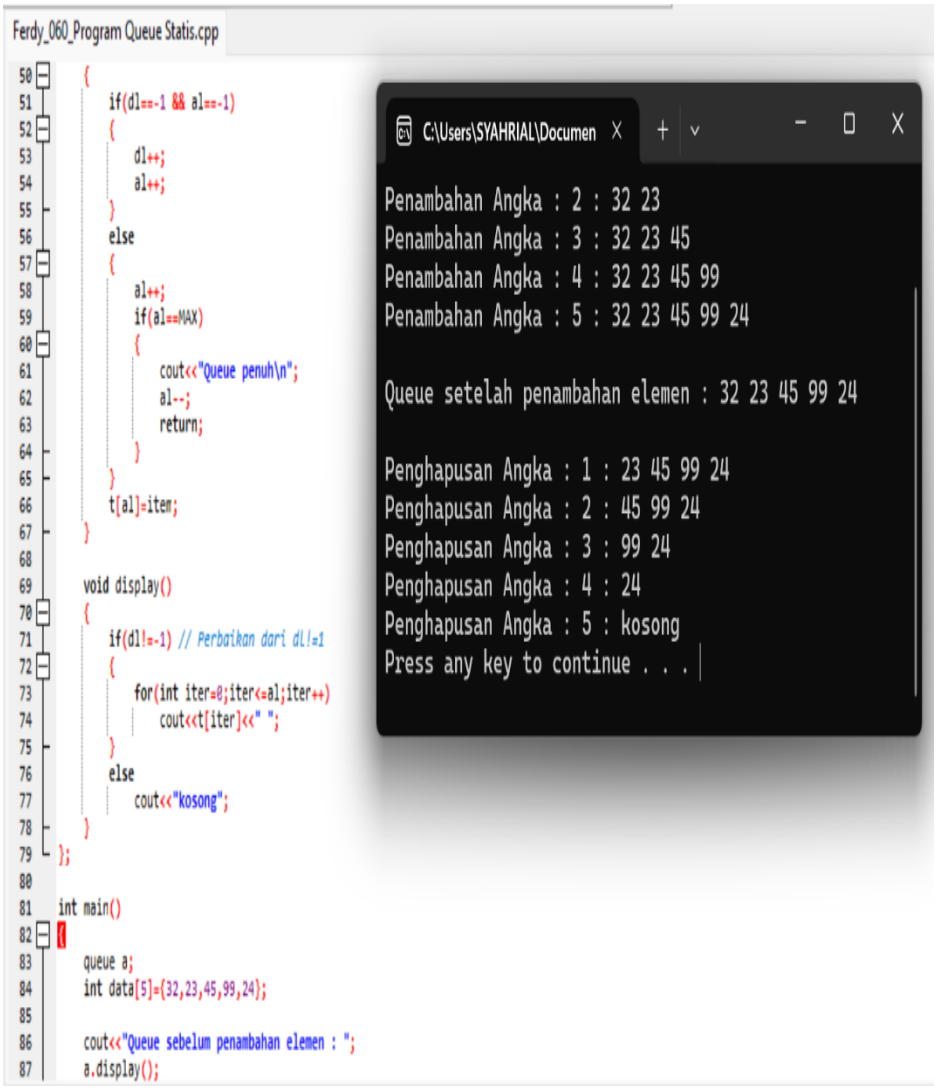

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
void add(int item)
    {
        if(dl==-1 && al==-1)
        {
            dl++;
            al++;
        }
        else
        {
            al++;
            if(al==MAX)
            {
                cout<<"Queue penuh\n";
                al--;
                return;
            }
        }
        t[al]=item;
    }

    void display()
    {
        if(dl!=-1) // Perbaikan dari dl!=1
        {
            for(int iter=0;iter<=al;iter++)
                cout<<t[iter]<<" ";
        }
        else
            cout<<"kosong";
    }
};

int main()
{
    queue a;
    int data[5]={32,23,45,99,24};

    cout<<"Queue sebelum penambahan elemen : ";
    a.display();
    cout<<endl<<endl;

    for(int iter=0;iter<5;iter++)
    {
        a.add(data[iter]);
        cout<<"Penambahan Angka : "<<(iter+1)<<" : ";
        a.display();
        cout<<endl;
    }

    cout<<endl;
    cout<<"Queue setelah penambahan elemen : ";
    a.display();
    cout<<endl<<endl;
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
for(int iter=0 ; iter<5 ; iter++)
    {
        a.del();
        cout<<"Penghapusan Angka : "<<(iter+1)<<" : ";
        a.display();
        cout<<endl;
    }

    system("pause");
    return 0;
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3. Buatlah program queue untuk memasukkan dan mengeluarkan data berupa huruf dalam antrian.

```
#include <iostream>
#include <cstdio>
#include <conio.h>
#include <cctype>

using namespace std;

#define MAXN 3

typedef char Titem;
typedef enum { NOT_OK, OK } Tboolean;

typedef struct {
    Titem array[MAXN];
    int first;
    int last;
    int number_of_items;
} Tqueue;

// Deklarasi fungsi
void initialize_queue(Tqueue *Pqueue);
Tboolean enqueue(Tqueue *p, Titem item);
Tboolean dequeue(Tqueue *p, Titem *Pitem);
void print_queue(const Tqueue *Pqueue);

int main() {
    Tqueue queue;
    Tboolean succeed;
    char chr;

    initialize_queue(&queue);

    printf("\n# Masukkan Sebuah Huruf Untuk Masuk Dalam
Antrian");
    printf("\n# Atau Tekan Angka 1 Untuk Mengeluarkan
Sebuah Huruf Dalam Antrian");
    printf("\n# Atau Tekan x Untuk Keluar : \n\n");

    // Mengambil input karakter pertama
    chr = getche();

    // Loop akan terus berjalan selama tidak menekan 'x',
'X', atau Enter
    while (chr != 'x' && chr != 'X' && chr != 10 && chr
!= 13) {

        if (chr == '1') {
            // Proses Dequeue (Mengeluarkan Antrian)
            succeed = dequeue(&queue, &chr);
            if (succeed) {
                printf("\n\n Huruf Yang Keluar Dari
Antrian adalah : %c", chr);
                print_queue(&queue);
            } else {
```

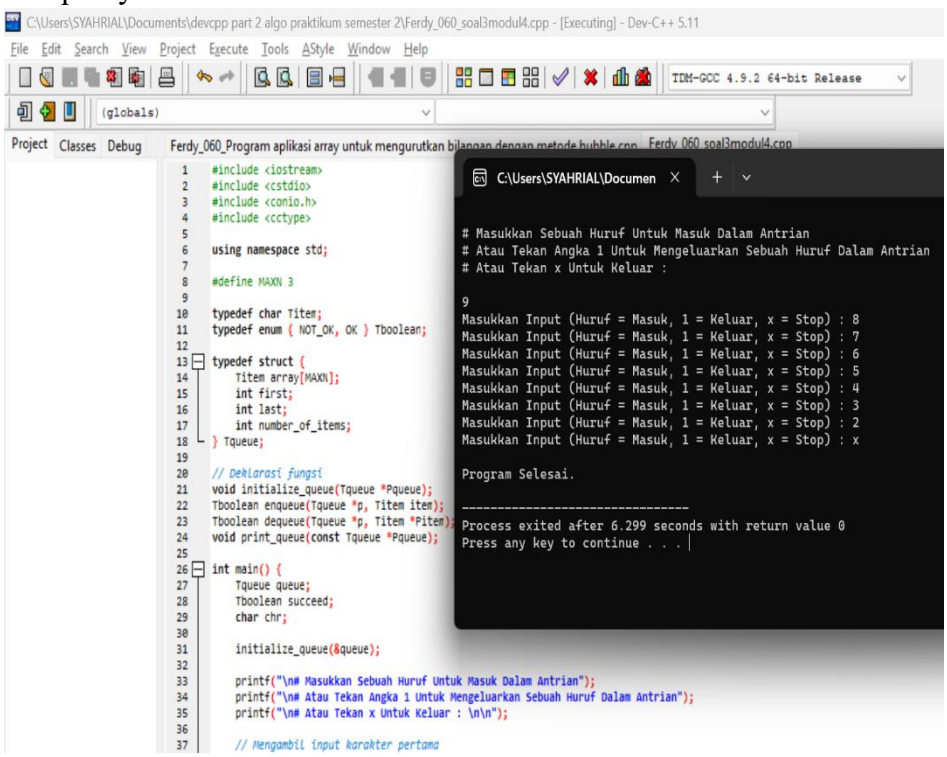

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
printf("\n\n Operasi Pengambilan Data Gagal (Antrian
Kosong)\n");
            }
        }
        else if (isalpha(chr)) {
            // Proses Enqueue (Memasukkan ke Antrian)
            succeed = enqueue(&queue, chr);
            if (!succeed) {
                printf("\n\n Operasi Pemasukan Data Gagal
(Antrian Penuh)\n");
            } else {
                print_queue(&queue);
            }
        }

        printf("\nMasukkan Input (Huruf = Masuk, 1 =
Keluar, x = Stop) : ");
        chr = getche();
    }

    printf("\n\nProgram Selesai.\n");
    return 0;
}

// Implementasi Fungsi

void initialize_queue(Tqueue *Pqueue) {
    Pqueue->first = 0;
    Pqueue->last = -1;
    Pqueue->number_of_items = 0;
}

Tboolean enqueue(Tqueue *Pqueue, Titem item) {
    if (Pqueue->number_of_items >= MAXN) {
        return NOT_OK;
    } else {
        Pqueue->last++;
        // Implementasi circular queue
        if (Pqueue->last > MAXN - 1) {
            Pqueue->last = 0;
        }
        Pqueue->array[Pqueue->last] = item;
        Pqueue->number_of_items++;
        return OK;
    }
}

Tboolean dequeue(Tqueue *Pqueue, Titem *Pitem) {
    if (Pqueue->number_of_items == 0) {
        return NOT_OK;
    } else {
        *Pitem = Pqueue->array[Pqueue->first++];
        // Implementasi circular queue
        if (Pqueue->first > MAXN - 1) {
            Pqueue->first = 0;
        }
        Pqueue->number_of_items--;
        return OK;
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
    }
}

void print_queue(const Tqueue *Pqueue) {
    int i, n;
    printf("\n Data Antrian Sekarang : ");

    if (Pqueue->number_of_items == 0) {
        printf("[Kosong]\n");
        return;
    }

    for (n = 1, i = Pqueue->first; n <= Pqueue->number_of_items; n++) {
        printf("%c ", Pqueue->array[i]);
        i++;
        if (i > MAXN - 1) {
            i = 0;
        }
    }
    printf("\n");
}
```

Outputnya

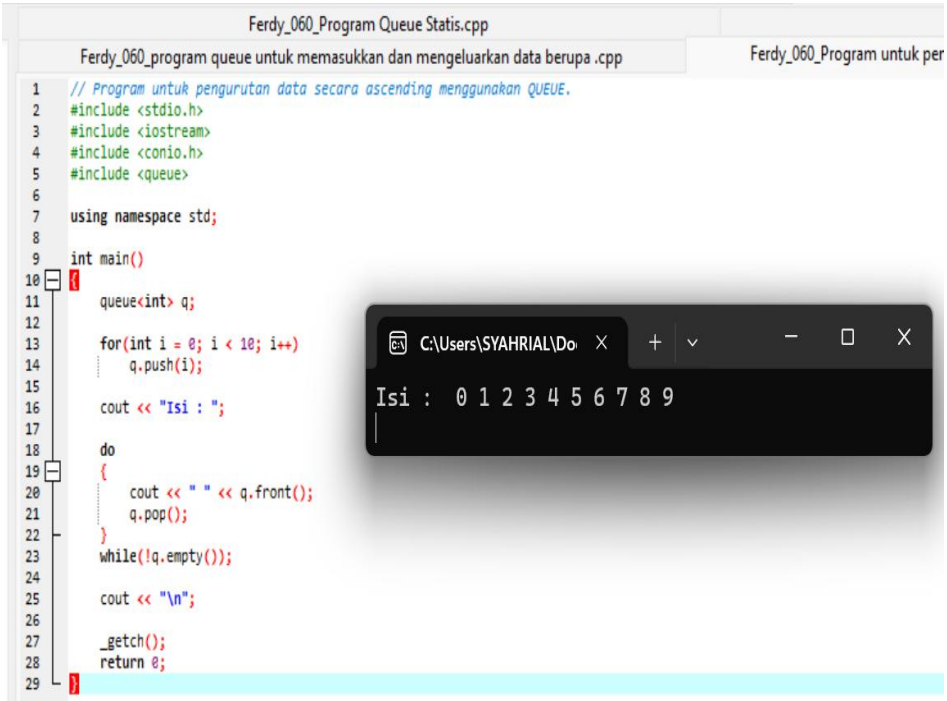


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

4. Program untuk pengurutan data secara ascending menggunakan QUEUE.

```
#include <stdio.h>
#include <iostream>
#include <conio.h>
#include <queue>

using namespace std;

int main()
{
    queue<int> q;

    for(int i = 0; i < 10; i++)
        q.push(i);

    cout << "Isi : ";

    do
    {
        cout << " " << q.front();
        q.pop();
    }
    while(!q.empty());

    cout << "\n";

    _getch();
    return 0;
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tugas!

1. Buatlah program queue untuk memasukkan dan mengeluarkan data berupa angka dalam antrian (Modifikasi program latihan 3).
2. Buatlah program untuk pengurutan data secara *Descending* menggunakan QUEUE (Modifikasi program latihan 4).

1. Buatlah program queue untuk memasukkan dan mengeluarkan data berupa angka dalam antrian.

```
#include <iostream>
#include <cstdio>
#include <cctype>
#include <conio.h>

using namespace std;

typedef char Titem;
#define MAXN 3

typedef enum {NOT_OK, OK} Tboolean;

typedef struct {
    Titem array[MAXN];
    int first;
    int last;
    int number_of_items;
} Tqueue;

void initialize_queue(Tqueue *Pqueue);
Tboolean enqueue(Tqueue *p, Titem item);
Tboolean dequeue(Tqueue *p, Titem *Pitem);
void print_queue(const Tqueue *Pqueue);

int main()
{
    Tqueue queue;
    Tboolean succeed;
    char chr;
    initialize_queue(&queue);

    printf("\n# Masukkan Sebuah Angka (0-9) untuk Masuk
Dalam Antrian \n");
    printf("\n# Atau Tekan Huruf L Untuk Mengeluarkan
Sebuah Angka Dalam Antrian \n");
    printf("\n# Atau Tekan X Untuk Keluar : \n\n");

    chr = getche();
    while(chr != 10 && chr != 13)
    {
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

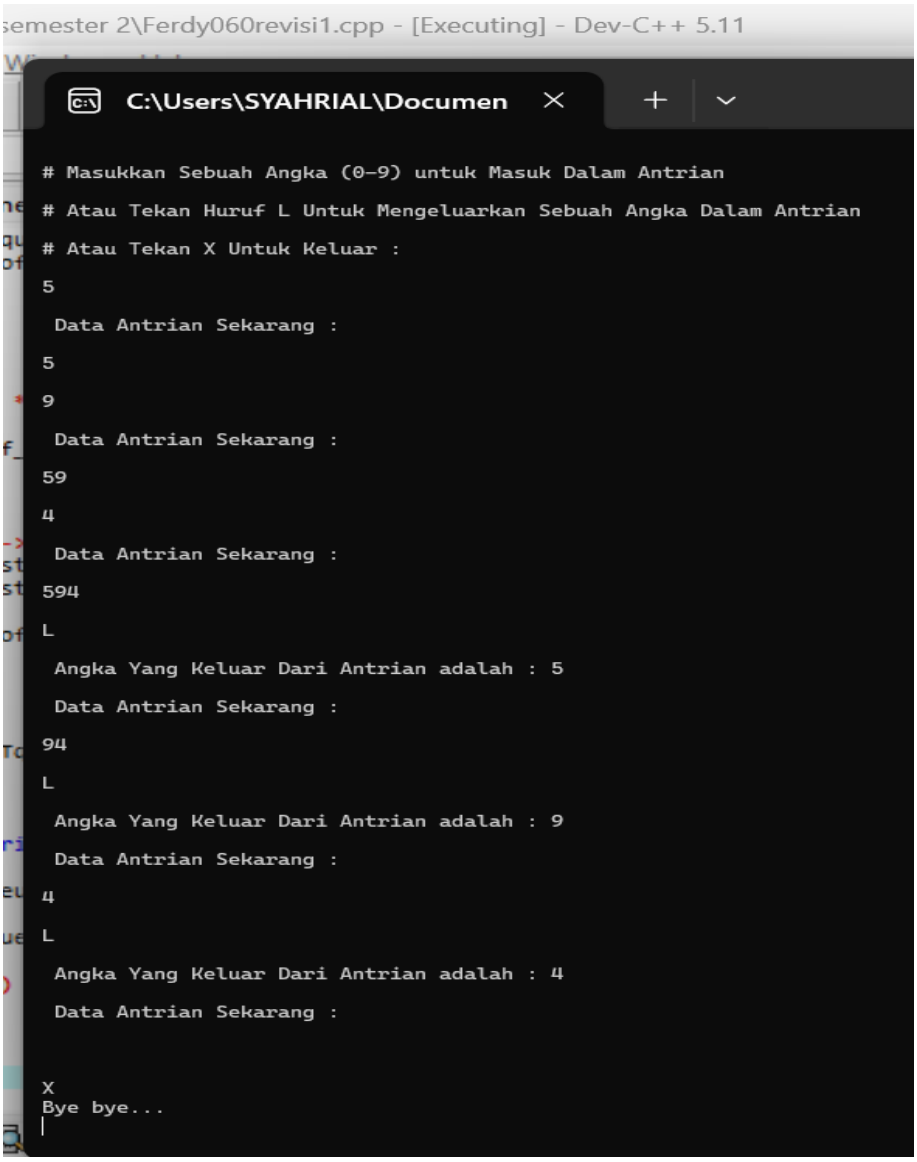
```
if (isalpha(chr))
        {
            // Sengaja dibiarkan kosong sesuai logika
asli untuk memisahkan angka dan huruf
        }
        else
        {
            succeed = enqueue(&queue, chr);
            print_queue(&queue);
            if (!succeed)
printf("\n Operasi Pemasukan Data Gagal \n");
        }

        // Memastikan program bisa membaca huruf 'l'
kecil maupun 'L' besar
        if(chr == 'l' || chr == 'L')
        {
            succeed = dequeue(&queue, &chr);
            if (succeed)
            {
                printf("\n");
                printf("\n Angka Yang Keluar Dari Antrian
adalah : %c", chr);
                print_queue(&queue);
            }
            else
                printf("\nOperasi    Pengambilan    Data
Gagal\n");
        }

        // Memastikan program bisa membaca huruf 'x'
kecil maupun 'X' besar
        if(chr == 'x' || chr == 'X')
        {
            printf("\nBye bye...\n");
            break;
        }

        chr = getche();
    }
    _getch();
    return 0;
}

void initialize_queue(Tqueue *Pqueue)
{
    Pqueue->first = 0;
    Pqueue->last = -1;
    Pqueue->number_of_items = 0;
}

Tboolean enqueue(Tqueue *Pqueue, Titem item)
{
    if (Pqueue->number_of_items >= MAXN)
        return(NOT_OK);
    else
    {
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
        Pqueue->last++;
        if (Pqueue->last > MAXN - 1)
            Pqueue->last = 0;

        Pqueue->array[Pqueue->last] = item;
        Pqueue->number_of_items++;
        return(OK);
    }
}


Tboolean dequeue(Tqueue *Pqueue, Titem *Pitem)
{
    if(Pqueue->number_of_items == 0)
        return(NOT_OK);
    else
    {
        *Pitem = Pqueue->array[Pqueue->first++];
        if (Pqueue->first > MAXN - 1)
            Pqueue->first = 0;

        Pqueue->number_of_items--;
        return(OK);
    }
}

void print_queue(const Tqueue *Pqueue)
{
    int i, n;
    printf("\n");
    printf("\n Data Antrian Sekarang : \n\n");

    for (n = 1, i = Pqueue->first; n <= Pqueue-
>number_of_items; n++)
    {
        printf("%c", Pqueue->array[i]);
        i++;
        if(i > MAXN - 1)
            i = 0;
    }
    printf("\n\n");
}
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Output

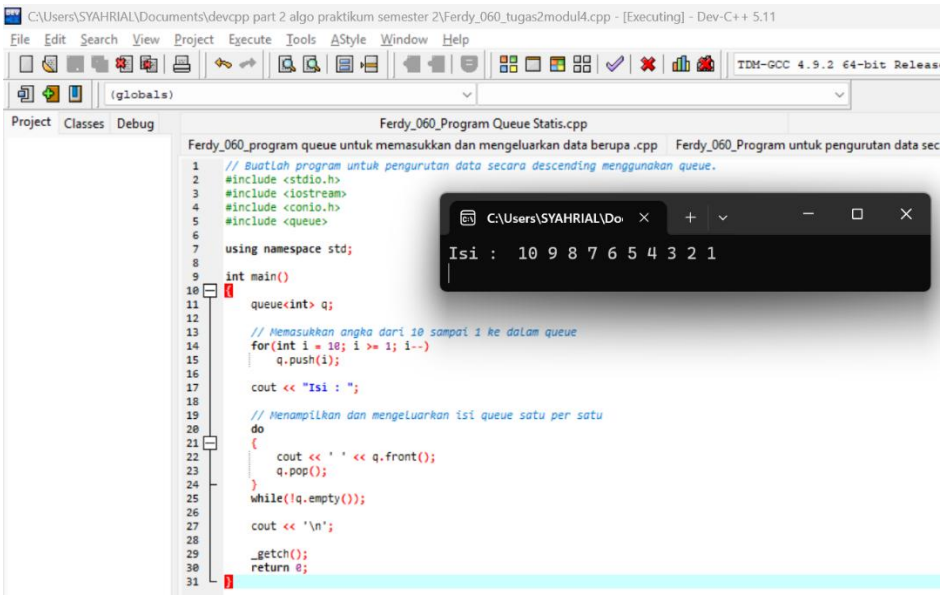


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

2. Buatlah program untuk pengurutan data secara descending menggunakan queue.

```
#include <stdio.h>
#include <iostream>
#include <conio.h>
#include <queue>

using namespace std;

int main()
{
    queue<int> q;

    // Memasukkan angka dari 10 sampai 1 ke dalam queue
    for(int i = 10; i >= 1; i--)
        q.push(i);

    cout << "Isi : ";

    // Menampilkan dan mengeluarkan isi queue satu per satu
    do
    {
        cout << ' ' << q.front();
        q.pop();
    }
    while(!q.empty());

    cout << '\n';

    _getch();
    return 0;
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 - 2026

Lembar Asistensi

Praktikum Algoritma dan Struktur Data
Pokok Bahasan 5

Judul	: Rekursif
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 30 Juni 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
(Yunianita Rahmawati, S.Kom. M.Kom)	**(Nadya Mulya Islami)**

POKOK BAHASAN 5

REKURSIF

A. PENDAHULUAN

Pokok bahasan ini membahas Rekursif dalam algoritma merujuk pada teknik di mana sebuah fungsi memanggil dirinya sendiri untuk memecahkan masalah yang lebih kecil atau lebih sederhana, sampai mencapai kondisi dasar yang menghentikan pemanggilan tersebut. Pada umumnya, rekursif digunakan untuk menyelesaikan masalah yang dapat dipecah menjadi sub-masalah yang serupa. Sebagai contoh, masalah faktorial atau deret Fibonacci sering diselesaikan menggunakan rekursif.

B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Mahasiswa mengetahui dan memahami definisi rekursif.
2. Mahasiswa memahami sifat-sifat rekursif.
3. Mahasiswa mengaplikasikan rekursif.

C. PENYAJIAN (TUTORIAL)

Fungsi rekursif adalah suatu fungsi yang memanggil dirinya sendiri, artinya fungsi tersebut dipanggil di dalam tubuh fungsi itu sendiri. Contoh menghitung nilai faktorial. Rekursif sangat memudahkan untuk memecahkan permasalahan yang kompleks. Sifat- sifat rekursif :

- Dapat digunakan ketika inti dari masalah terjadi berulang kali.
- Sedikit lebih efisien dari iterasi tapi lebih elegan.
- Method-methodnya dimungkinkan untuk memanggil dirinya sendiri.

Data yang berada dalam method tersebut seperti argument disimpan sementara ke dalam stack sampai method pemanggil diselesaikan.

Contoh sederhana rekursif dalam algoritma adalah menghitung faktorial dari suatu angka:

```
def faktorial(n):
    if n == 0:
      return 1
    else:
      return n * faktorial(n - 1)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Penjelasan:

- Fungsi faktorial akan memanggil dirinya sendiri dengan nilai yang lebih kecil (n-1).
- Ketika n mencapai 0, fungsi akan berhenti dan mengembalikan nilai 1 (ini adalah kondisi dasar).
- Setiap pemanggilan fungsi mengalikan n dengan hasil faktorial dari n-1.

Secara umum, setiap rekursi membutuhkan dua hal penting:

1. Kondisi yang menentukan kapan rekursi berhenti.
2. Pemanggilan fungsi yang lebih kecil dari masalah utama.

Penggunaan rekursif harus dilakukan hati-hati, karena jika tidak ada kondisi dasar atau jika pemanggilan rekursif terlalu dalam, dapat menyebabkan stack overflow atau masalah performa

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

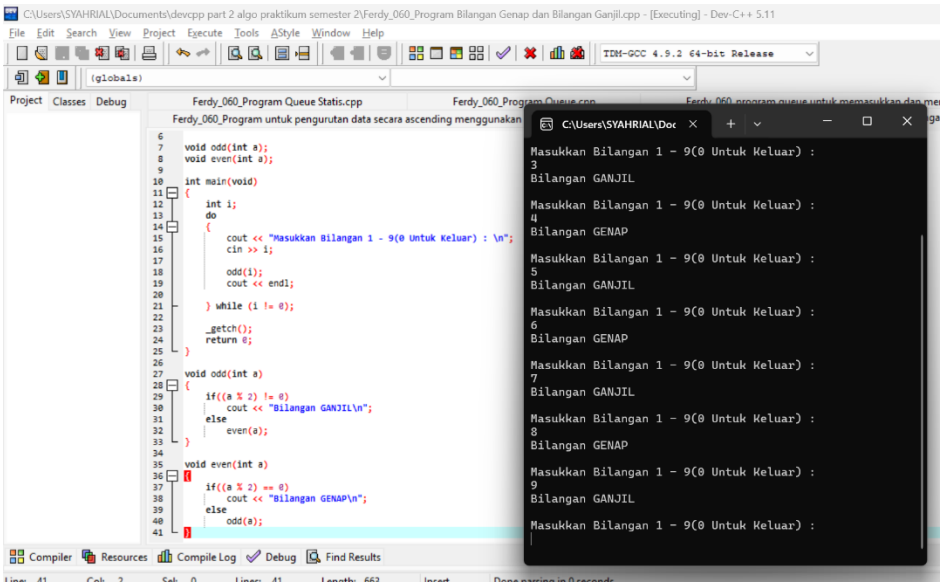
LEMBAR KERJA DAN TUGAS

1. Program Bilangan Genap dan Bilangan Ganjil.

```
#include <iostream>
#include <conio.h>

using namespace std;

void odd(int a);
void even(int a);

int main(void)
{
    int i;
    do
    {
        cout << "Masukkan Bilangan 1 - 9(0 Untuk Keluar)
: \n";
        cin >> i;

        odd(i);
        cout << endl;

    } while (i != 0);

    _getch();
    return 0;
}

void odd(int a)
{
    if((a % 2) != 0)
        cout << "Bilangan GANJIL\n";
    else
        even(a);
}

void even(int a)
{
    if((a % 2) == 0)
        cout << "Bilangan GENAP\n";
    else
        odd(a);
}
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Output :

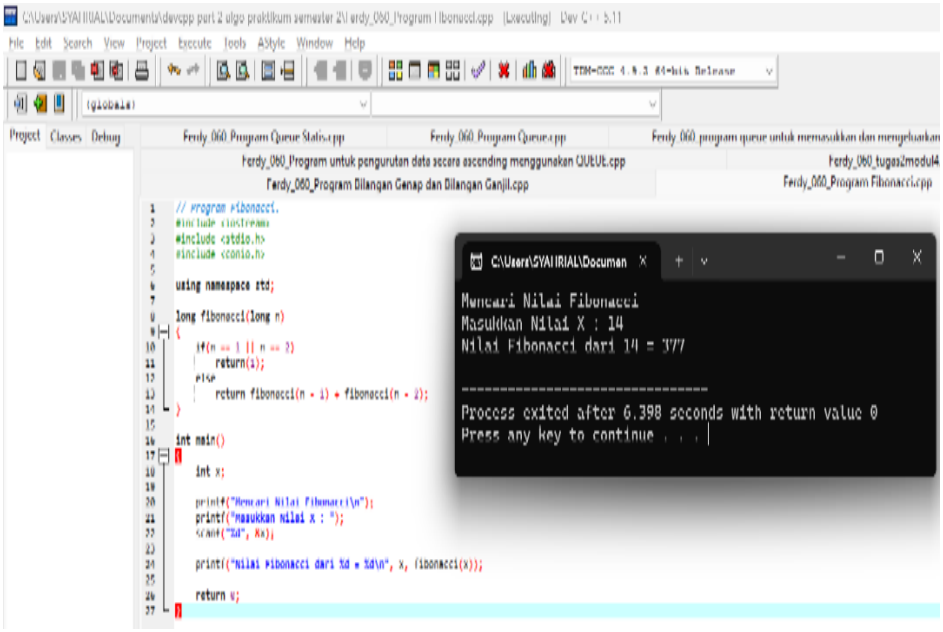


2. Program Fibonacci.

```
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

long fibonacci(long n)
{
    if(n == 1 || n == 2)
        return(1);
    else
        return fibonacci(n - 1) + fibonacci(n - 2);
}

int main()
{
    int x;

    printf("Mencari Nilai Fibonacci\n");
    printf("Masukkan Nilai X : ");
    scanf("%d", &x);

    printf("Nilai Fibonacci dari %d = %d\n", x,
fibonacci(x));

    return 0;
}
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Output :

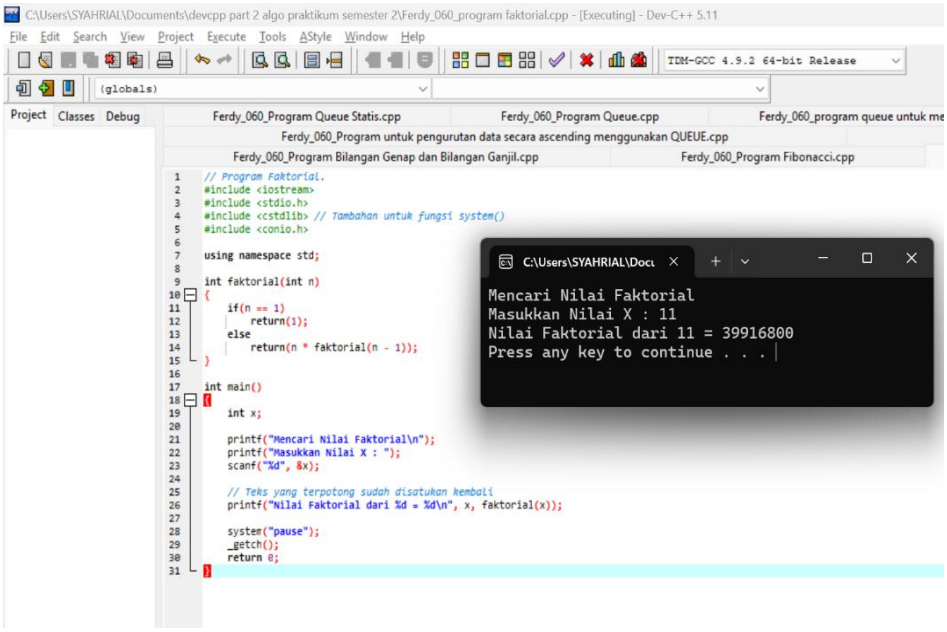


3. Program Faktorial.

```
#include <iostream>
#include <stdio.h>
#include <cstdlib> // Tambahan untuk fungsi system()
#include <conio.h>

using namespace std;

int faktorial(int n)
{
    if(n == 1)
        return(1);
    else
        return(n * faktorial(n - 1));
}

int main()
{
    int x;

    printf("Mencari Nilai Faktorial\n");
    printf("Masukkan Nilai X : ");
    scanf("%d", &x);

    // Teks yang terpotong sudah disatukan kembali
    printf("Nilai Faktorial dari %d = %d\n", x, faktorial(x));

    system("pause");
    _getch();
    return 0;
}
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Output :



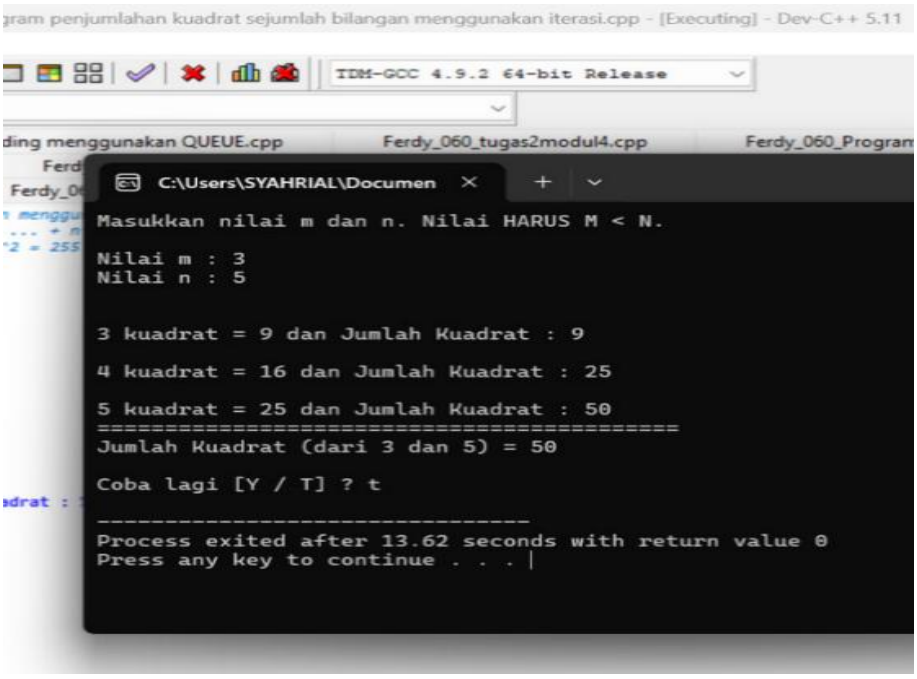
4. Program penjumlahan kuadrat sejumlah bilangan menggunakan iterasi. *jumKuadrat (m,n) = $m^2 + (m+1)^2 + (m+2)^2 + \ldots + n^2$*
dengan m<n. jumKuadrat (5,9) =$5^2 + 6^2 + 7^2 + 8^2 + 9^2 = 255$

```
#include <iostream>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

using namespace std;

int jumkuadrat(int m, int n)
{
    int i, jum=0;
    for(i=m; i<=n; i++)
    {
        jum += i*i;
        printf("\n%d kuadrat = %d dan Jumlah Kuadrat :
%d\n", i, i*i, jum);
    }
    return jum;
}

int main(void)
{
    char cNilaim[3], cNilain[3], cJawab[2];
    int iNilaim, iNilain, iNilaijumkuadrat;

    while(1)
    {
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
printf("Masukkan nilai m dan n. Nilai HARUS M < N.\n");
        cout << endl;

        printf("Nilai m : ");
        cin >> cNilaim; // Mengganti gets() agar tidak
error di g++ modern

        printf("Nilai n : ");
        cin >> cNilain;

        cout << endl;

        iNilaim = atoi(cNilaim);
        iNilain = atoi(cNilain);

        iNilaijumkuadrat = jumkuadrat(iNilaim, iNilain);

        cout                                              <<
"=========================================";
        printf("\nJumlah Kuadrat (dari %d dan %d) = %d",
iNilaim, iNilain, iNilaijumkuadrat);
        cout << endl;

        printf("\nCoba lagi [Y / T] ? ");
        cin >> cJawab;
        if(strcmp(cJawab,"t")==0                          ||
strcmp(cJawab,"T")==0)
            break;
    }
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tugas !

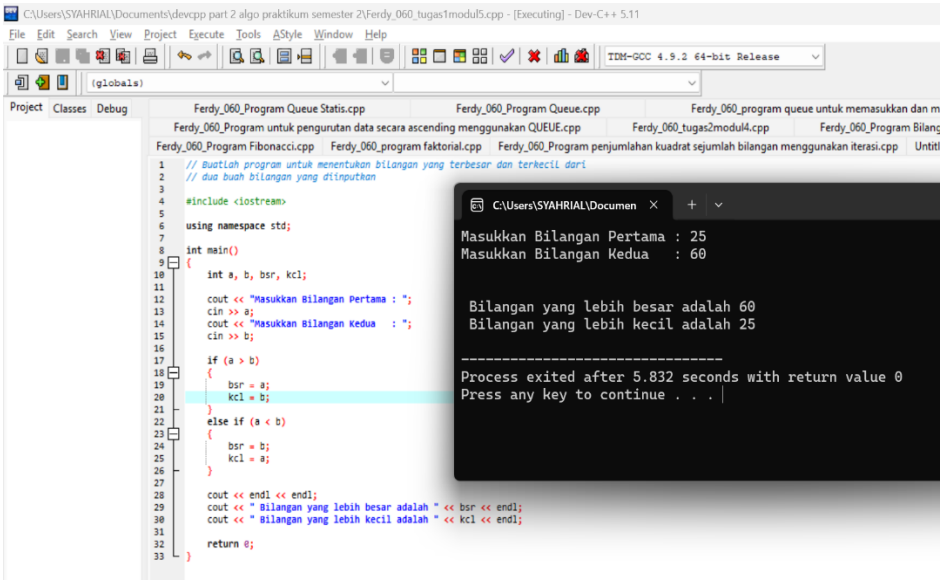
1. Buatlah program untuk menetukan bilangan yang terbesar dan terkecil dari dua buah bilangan yang diinputkan.
2. Buatlah program dengan cara rekursi untuk menampilkan perkalian 3 buah bilangan, dimana ketiga bilangan tersebut nilainya diinputkan.
3. Buatlah program untuk mengurutkan bilangan secara *Descensding* dengan menggunakan *Rekursif.*
4. Buatlah program perhitungan gaji karyawan. Dengan ketentuan :
   a. Jika gaji kotor <= 1.500.000 maka mendapat potongan 10%.
   b. Jika gaji kotor > 1.500.000 maka mendapat potongan 15%.

1. Buatlah program untuk menentukan bilangan yang terbesar dan terkecil dari dua buah bilangan yang diinputkan

```
#include <iostream>

using namespace std;

int main()
{
    int a, b, bsr, kcl;

    cout << "Masukkan Bilangan Pertama : ";
    cin >> a;
    cout << "Masukkan Bilangan Kedua   : ";
    cin >> b;

    if (a > b)
    {
        bsr = a;
        kcl = b;
    }
    else if (a < b)
    {
        bsr = b;
        kcl = a;
    }

    cout << endl << endl;
    cout << " Bilangan yang lebih besar adalah " << bsr
<< endl;
    cout << " Bilangan yang lebih kecil adalah " << kcl
<< endl;

    return 0;
}
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Output :



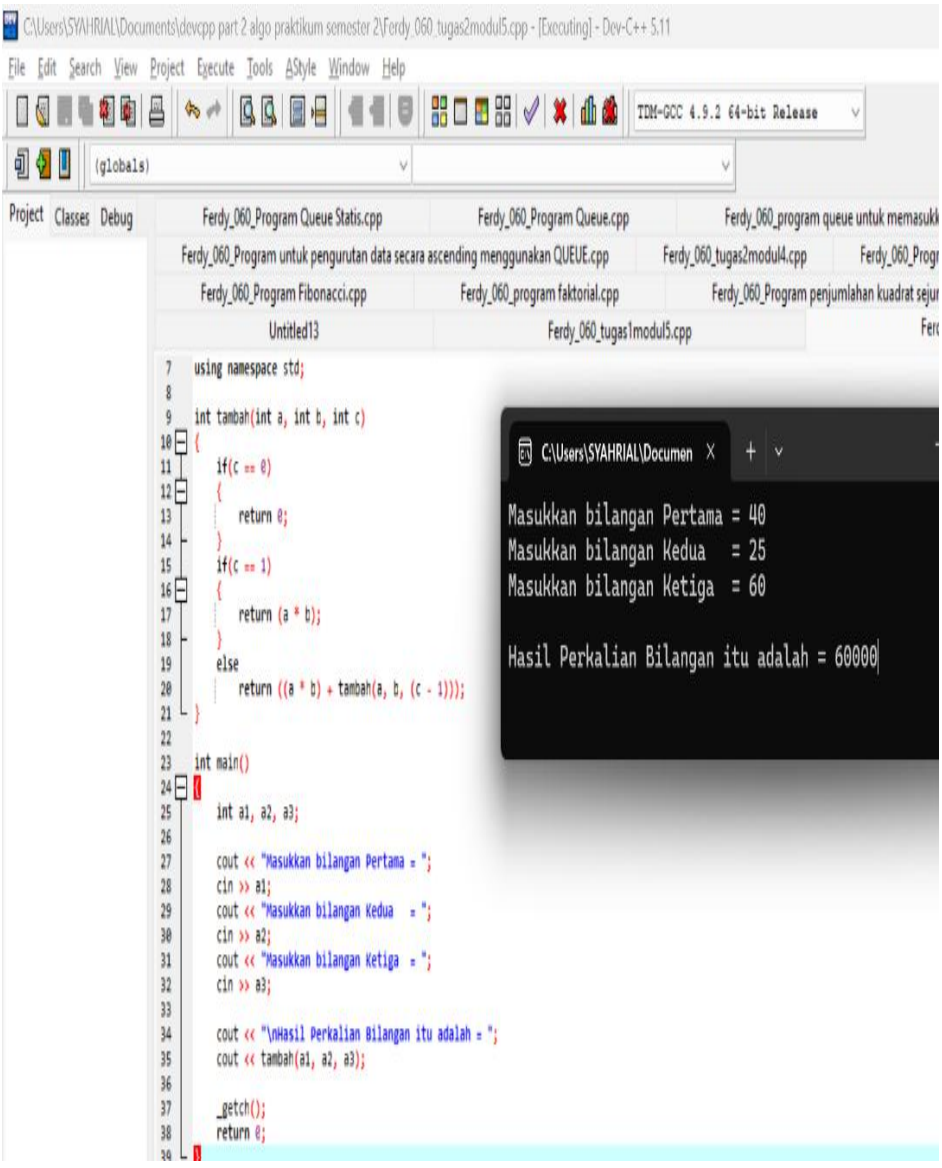
2. Buatlah program dengan cara rekursi untuk menampilkan 3 buah bilangan, dimana ketiga bilangan tersebut nilainya diinputkan.

```
#include <iostream>
#include <conio.h>

using namespace std;

int tambah(int a, int b, int c)
{
    if(c == 0)
    {
        return 0;
    }
    if(c == 1)
    {
        return (a * b);
    }
    else
        return ((a * b) + tambah(a, b, (c - 1)));
}

int main()
{
    int a1, a2, a3;

    cout << "Masukkan bilangan Pertama = ";
    cin >> a1;
    cout << "Masukkan bilangan Kedua   = ";
    cin >> a2;
    cout << "Masukkan bilangan Ketiga  = ";
    cin >> a3;

    cout << "\nHasil Perkalian Bilangan itu adalah = ";
    cout << tambah(a1, a2, a3);
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
    _getch();
    return 0;
}
```

Output :



Program ini menggunakan #include <conio.h> dan getch() yang keduanya berkaitan dengan input/output konsol yang spesifik untuk sistem operasi Windows dan mungkin tidak berfungsi di semua lingkungan pengembangan

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

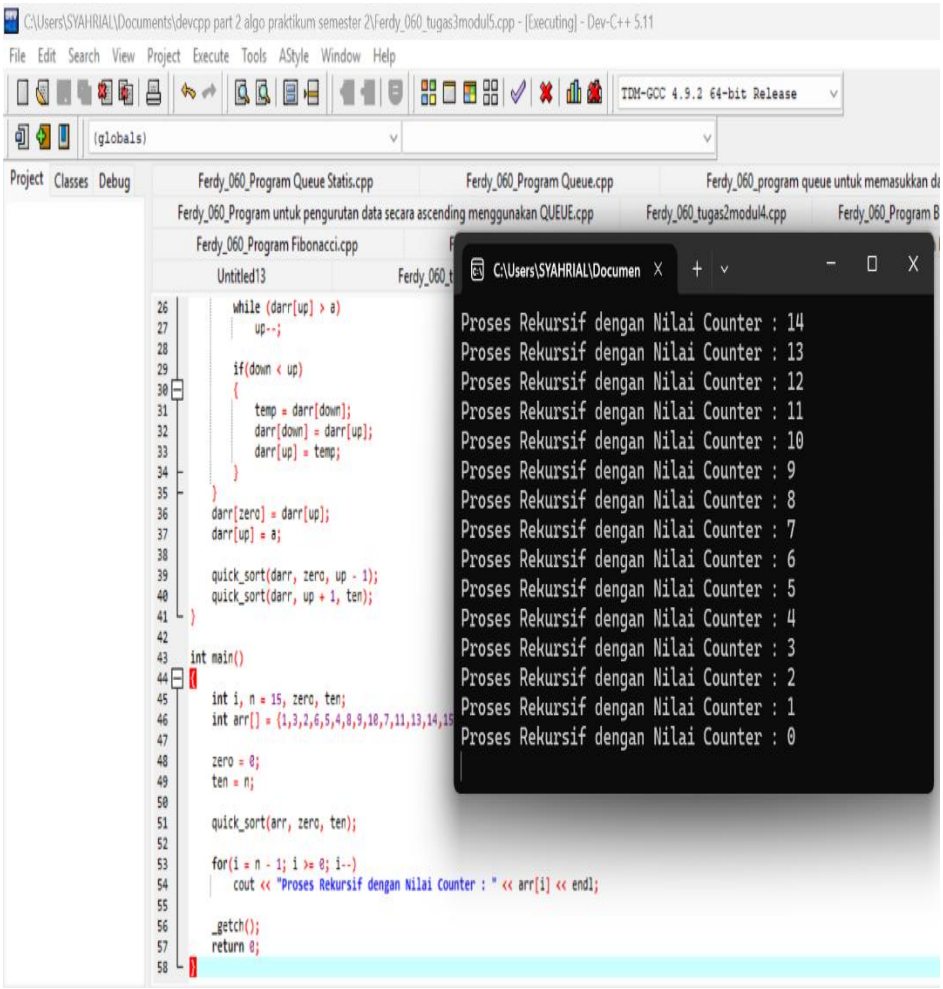
3. Buatlah program untuk mengurutkan bilangan secara descending dengan menggunakan rekursif.

```
#include <iostream>
#include <conio.h>
#define max 20

using namespace std;

void quick_sort(int darr[max], int zero, int ten)
{
    int a;
    int up, down;
    int temp;

    if (zero >= ten)
        return;

    a = darr[zero];
    up = ten;
    down = zero;

    while (down < up)
    {
        while (darr[down] <= a)
            down++;
        while (darr[up] > a)
            up--;

        if(down < up)
        {
            temp = darr[down];
            darr[down] = darr[up];
            darr[up] = temp;
        }
    }
    darr[zero] = darr[up];
    darr[up] = a;

    quick_sort(darr, zero, up - 1);
    quick_sort(darr, up + 1, ten);
}

int main()
{
    int i, n = 15, zero, ten;
    int arr[] = {1,3,2,6,5,4,8,9,10,7,11,13,14,15,12};

    zero = 0;
    ten = n;

    quick_sort(arr, zero, ten);
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
for(i = n - 1; i >= 0; i--)
        cout << "Proses Rekursif dengan Nilai Counter :
" << arr[i] << endl;

    _getch();
    return 0;
}
```

Outputnya



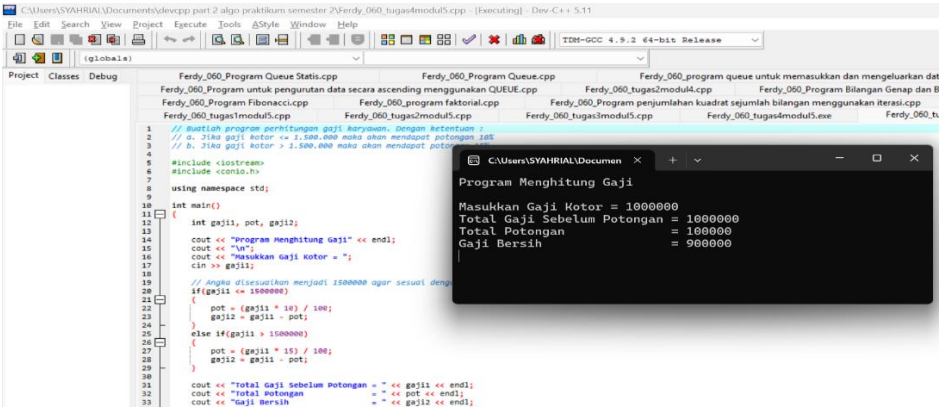
Program ini menggunakan #include <conio.h> dan getch() yang keduanya berkaitan dengan input/output konsol yang spesifik untuk sistem operasi Windows dan mungkin tidak berfungsi di semua lingkungan pengembangan.

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

4. Buatlah program perhitungan gaji karyawan. Dengan ketentuan :
   a. Jika gaji kotor <= 1.500.000 maka akan mendapat potongan 10%
   b. Jika gaji kotor > 1.500.000 maka akan mendapat potongan 15%

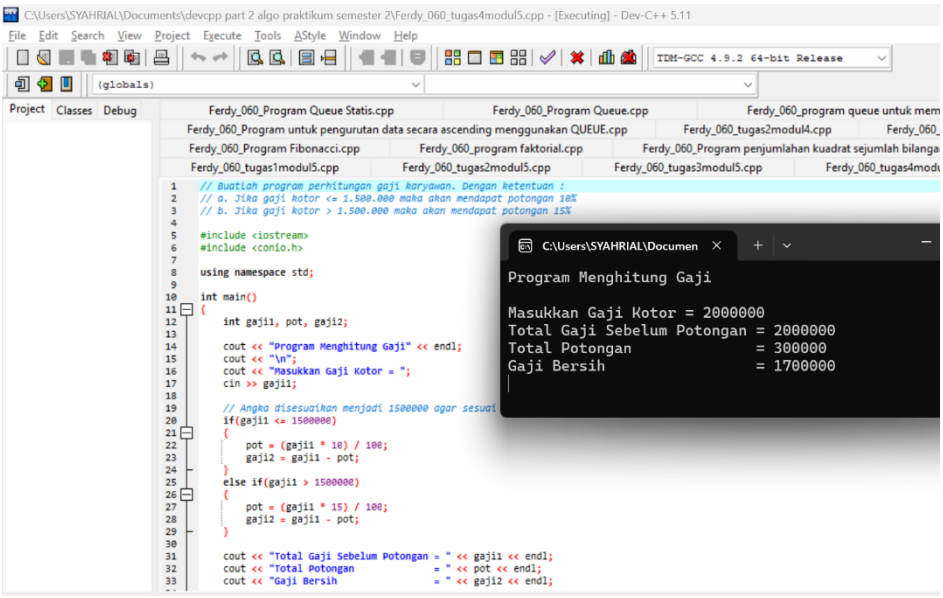
```
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    int gaji1, pot, gaji2;

    cout << "Program Menghitung Gaji" << endl;
    cout << "\n";
    cout << "Masukkan Gaji Kotor = ";
    cin >> gaji1;

    // Angka disesuaikan menjadi 1500000 agar sesuai
dengan ketentuan soal
    if(gaji1 <= 1500000)
    {
        pot = (gaji1 * 10) / 100;
        gaji2 = gaji1 - pot;
    }
    else if(gaji1 > 1500000)
    {
pot = (gaji1 * 15) / 100;
        gaji2 = gaji1 - pot;
    }

    cout << "Total Gaji Sebelum Potongan = " << gaji1 <<
endl;
    cout << "Total Potongan                = " << pot <<
endl;
    cout << "Gaji Bersih                   = " << gaji2 <<
endl;

    _getch();
    return 0;
}
```

Output Gaji Kotor <= 1.500.000



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Output Gaji Kotor > 1.500.000



Dengan menjalankan program ini, Anda dapat menghitung gaji bersih berdasarkan gaji kotor dengan menerapkan potongan yang sesuai. Program akan mencetak total gaji sebelum potongan, total potongan, dan gaji bersih setelah potongan.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 - 2026

Lembar Asistensi

Praktikum Algoritma dan Struktur Data

Pokok Bahasan 6

Judul	: Sorting (Pengurutan)
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 30 Juni 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
(Yunianita Rahmawati, S.Kom. M.Kom)	**(Nadya Mulya Islami)**

POKOK BAHASAN 6

SORTING (Pengurutan)

A. PENDAHULUAN

Pokok bahasan ini akan dibahas mengenai Sorting (pengurutan) adalah proses untuk menyusun elemen-elemen dalam suatu kumpulan data, seperti array atau list, dalam urutan tertentu. Urutan ini bisa berdasarkan nilai elemen, baik secara menaik (ascending) atau menurun (descending). Tujuan utama dari sorting adalah untuk memudahkan pencarian, pengolahan data, atau analisis dengan cara menyusun data secara teratur.

B. TUJUAN

Setelah mempelajari bab ini diharapkan mahasiswa mampu :

1. Menunjukkan beberapa algoritma dalam pengurutan.
2. Menunjukkan bahwa pengurutan merupakan suatu persoalan yang bisa diselesaikan dengan sejumlah algoritma yang berbeda satu sama lain.
3. Dapat memilih algoritma yang paling sesuai untuk menyelesaikan suatu permasalahan pemrograman.

C. PENYAJAIAN (TUTORIAL)

Pengurutan data *(sorting)* didefinisikan sebagai suatu proses untuk menyusun kembali himpunan objek menggunakan aturan tertentu. Ada dua macam urutan yang bisa digunakan dalam proses pengurutan yaitu :

- Urutan naik *(ascending)* yaitu dari data yang mempunyai nilai paling kecil sampai paling besar.
- Urutan turun *(descending)* yaitu dari data yang mempunyai nilai paling besar sampai paling kecil.

Contoh : data bilangan 5, 2, 6, dan 4 dapat diurutkan naik menjadi 2, 4, 5, 6 atau diurutkan menjadi 6, 5, 4, 2. Pada data yang bertipe char, nilai data dikatakan lebih kecil atau lebih besar dari yang lain didasarkan pada urutan relative *(collating sequence)* seperti dinyatakan dalam tabel ASCII. Keuntungan dari data yang sudah dalam keadaan terurut yaitu :

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Data mudah dicari, mudah untuk dibetulkan, dihapus, disisipi atau digabungkan. Dalam keadaan terurutkan, kita mudah melakukan pengecekan apakah ada data yang hilang. Misalnya kamus Bahasa, buku telepon. Mempercepat proses pencarian data yang harus dilakukan berulang kali.

Beberapa faktor yang berpengaruh pada efektifitas suatu algoritma pengurutan antara lain :

- Banyak data yang diurutkan.
- Kapasitas pengingat apakah mampu menyimpan semua data yang kita miliki.
- Tempat penyimpanan data, misalnya piringan, pita tau kartu, dll.

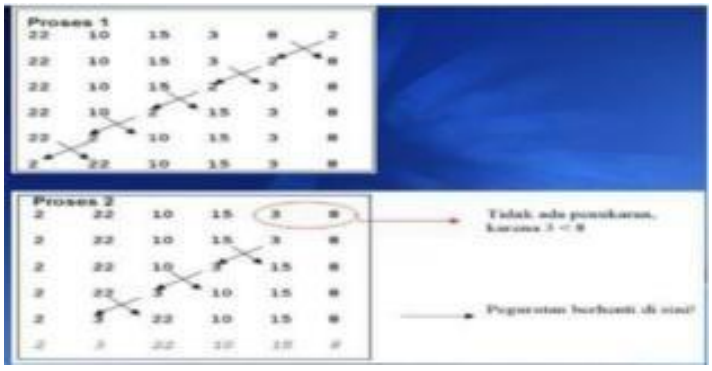
Beberapa algoritma metode pengurutan dan prosedurnya sebagai berikut :

1. *Bubble Sort*

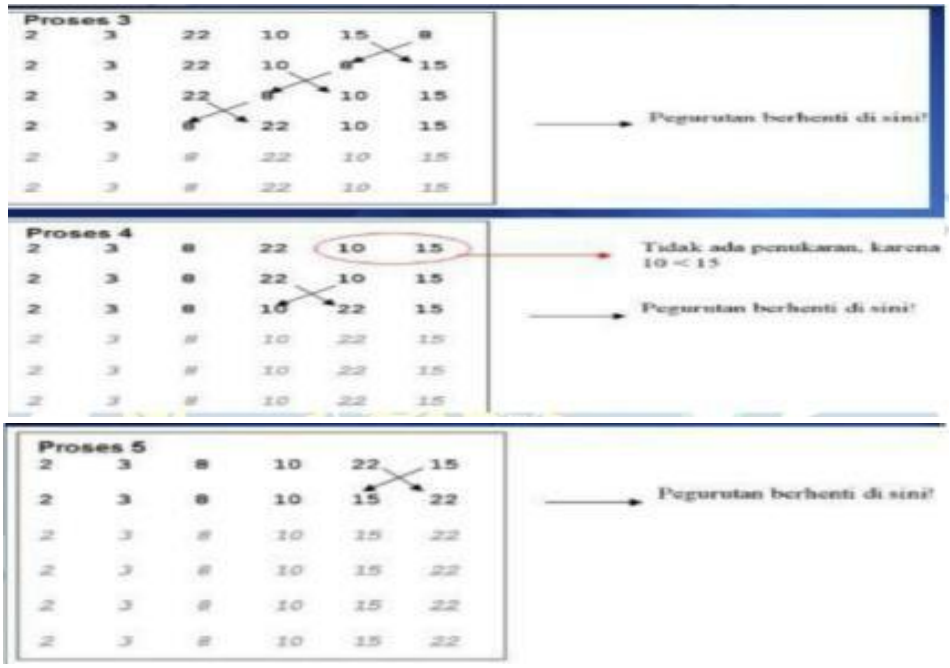
   Bubble Sort adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya. Apabila elemen sekarang > elemen berikutnya, maka posisinya ditukar. Kalua tidak, tidak perlu ditukar. Diberi nama "Bubble" karena proses pengurutan secara berangsur-angsur bergerak/berpindah ke posisinya yang tepat, seperti gelembung yang keluar dari sebuah gelas bersoda.

   Proses Bubble Sort:

   Data paling akhir dibandingkan dengan data di depanya, jika ternyata lebih kecil atau besar maka tukar sesuai dengan kekuatan (descending dan ascending). Dan pengecekan yang sama dilakukan terhadap data yang selanjutnya sampai data yang paling awal.



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271



Algoritma *Bubble Sort* :

1. i=0
2. selama (i<N-1) kerjakan baris 3 sampai 7
3. j=N-1
4. selama (j>=i) kerjakan baris 5 sampai 7
5. jika (Data[j-1] > Data[j]) maka tukar Data[j-1] dengan Data[j]
6. j=j-1
7. i=i+1

prosedur yang menggunakan metode gelembung:

```
void BubbleSort()
{


int i, j;

for(i=1; i=i; j--)
    if(Data[j-1] > Data[j])
        Tukar(&Data[j-1], &Data[j]);
}
```

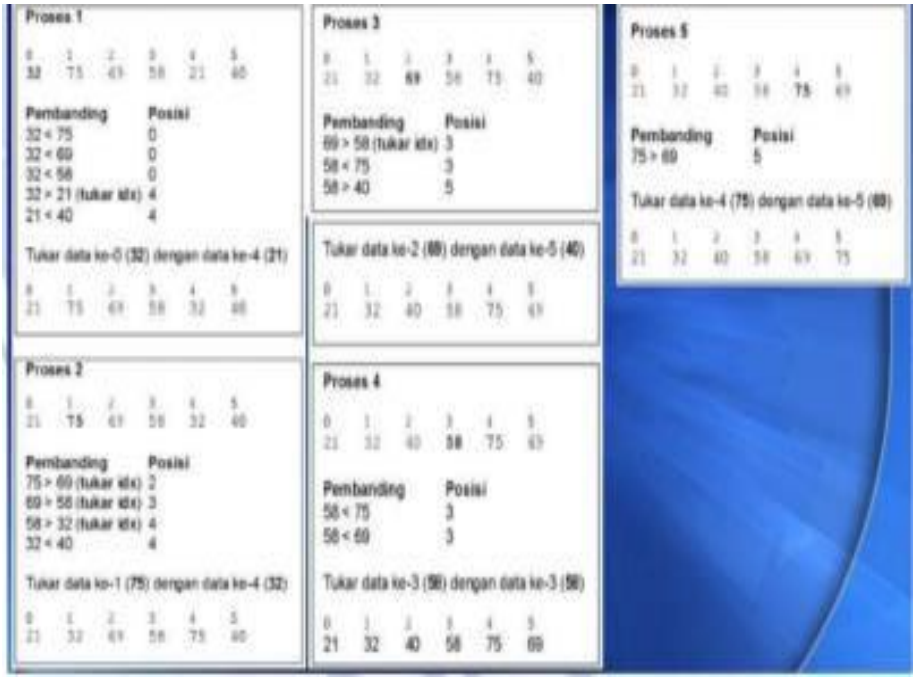

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

2. *Selection Sort*

Metode seleksi melakukan pengurutan dengan cara mencari data yang terkecil kemudian menukarnya dengan data yang digunakan sebagai acuan atau sering dinamakan pivot. Selama proses, pembandingan dan pengubahan hanya dilakukan pada indeks pembanding saja, pertukaran data secara fisik terjadi pada akhir proses. Proses pengurutan dengan metode seleksi dapat dijelaskan sebagai berikut :

Langkah pertama dicari data terkecil dari data pertama sampai data terakhir. Kemudian data terkecil ditukar dengan data pertama. Dengan demikian, data pertama sekarang mempunyai nilai paling kecil dibanding data yang lainnya.

Langkah kedua, data terkecil kita cari mulai dari data kedua sampai terakhir. Data terkecil yang kita peroleh ditukar dengan data kedua dan demikian seterusnya sampai semua elemen dalam keadaan terurutkan.



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Algoritma seleksi dapat dituliskan sebagai berikut :

```
1. i = 0
2. selama (i < N-1) kerjakan baris 3 sampai dengan 9
3. k = i
4. j = i + 1
5. Selama (j < N) kerjakan baris 6 dan 7
6. Jika (Data[k] > Data[j]) maka k = j
7. j = j + 1
8. Tukar Data[i] dengan Data[k]
9. i = i + 1
```

Di bawah ini merupakan prosedur yang menggunakan metode seleksi :

```
void SelectionSort()
{
    int i, j, k;

    for(i=0; i<Max-1;i++)
    {
        k = i;

        for (j=i+1; j<Max; j++)
            if(Data[k] > Data[j])
                k = j;

        Tukar(&Data[i], &Data[k]);
    }
}
```

3. *Merge Sort*

Algoritma Merge Sort adalah algoritma pengurutan yang berdasarkan pada strategi *divide* and *conquer*. Algoritma ini terdiri dari dua bagian utama, pembagian list yang diberikan untuk di-sort ke dalam beberapa sublist yang lebih kecil, dan sort (mengurutkan) dan merge (menggabungkan) *sublist* - *sublist* yang lebih kecil ke dalam list yang sudah diurutkan. Pembagian bisa dikatakan cukup mudah karena sublist-sublist tersebut dibagi kedalam dua sublist yang ukuranya adalah setengah dari ukuran semula. Hal ini terus diulang sampai sublist itu cukup kecil untuk di-*sort* secara efisien (umumnya telah terdiri dari satu atau dua elemen). Dalam langkah merge dua *sublist* disatukan kembali dan diurutkan pada saat yang sama. Algoritma untuk *merge sort* adalah sebagai berikut :

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

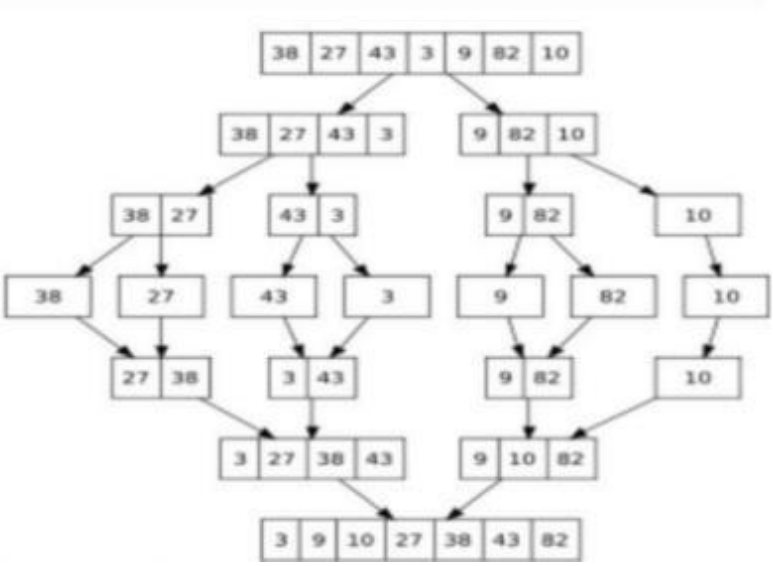
A. Untuk kasus n=1, maka tab a sudah terurut sendirinya (langkah solve)

B. Untuk kasus n>1, maka :

   a. DIVINE: bagi table a menjadi dua bagian, bagian kiri dan bagian kanan, masing-masing bagian berukuran n/2 elemen.

   b. CONOQUER: secara rekursif, terapkan algoritma D-and-C pada masing-masing bagian.

   c. MERGE: gabung hasil pengurutan kedua bagian sehingga diperoleh table a yang terurut.



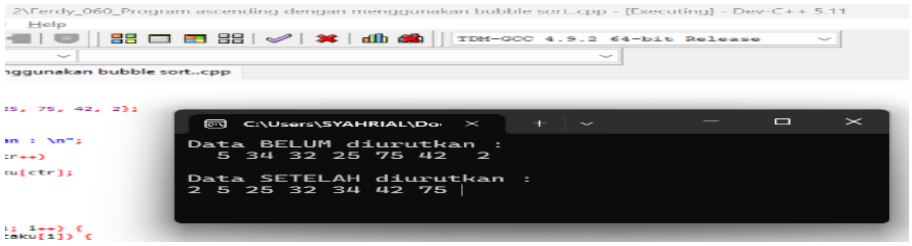
fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

1. Program *ascending* dengan menggunakan *bubble sort.*

```
#include <iostream>
#include <conio.h>
#include <iomanip>

using namespace std;

int main(void)
{
    int dataku[] = {5, 34, 32, 25, 75, 42, 2};
    int adaPertukaran;
    int n;

    cout << "Data BELUM diurutkan : \n";

    for(int ctr = 0; ctr < 7; ctr++)
    {
        cout << setw(3) << dataku[ctr];
    }
    cout << endl << endl;

    do {
        adaPertukaran = 0;

        for(int i = 0; i < 7 - 1; i++) {
            if(dataku[i+1] < dataku[i]) {
                n = dataku[i];
                dataku[i] = dataku[i+1];
                dataku[i+1] = n;
                adaPertukaran = 1;
            }
        }
    } while(adaPertukaran == 1);

    cout << "Data SETELAH diurutkan : \n";
    for(int i = 0; i < 7; i++) {
        cout << dataku[i];
        cout << " ";
    }

    _getch();
    return 0;
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

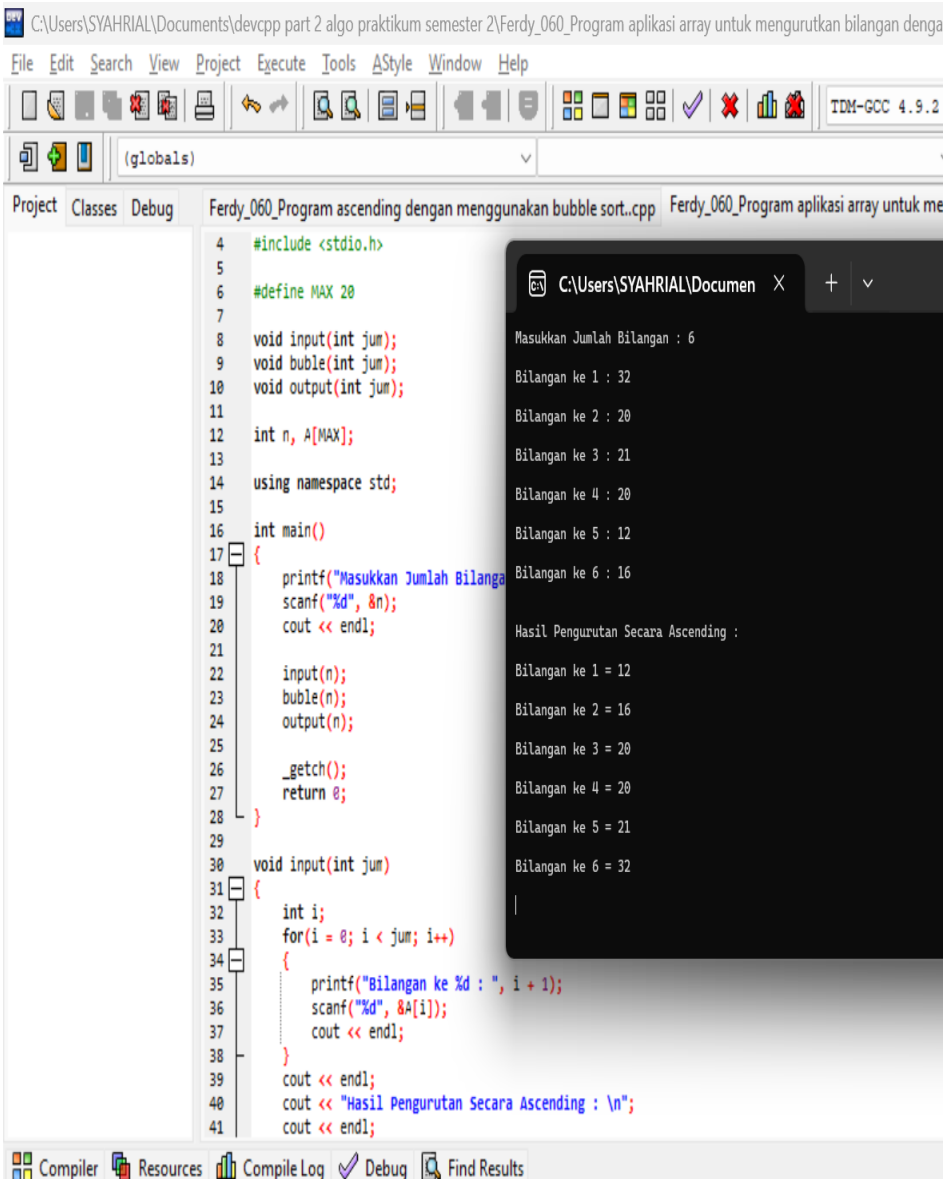
2. Program aplikasi array untuk mengurutkan bilangan dengan metode *bubble sort.*

```
#include <iostream>
#include <conio.h>
#include <stdio.h>

#define MAX 20

void input(int jum);
void buble(int jum);
void output(int jum);

int n, A[MAX];

using namespace std;

int main()
{
    printf("Masukkan Jumlah Bilangan : ");
    scanf("%d", &n);
    cout << endl;

    input(n);
    buble(n);
    output(n);

    _getch();
    return 0;
}

void input(int jum)
{
    int i;
    for(i = 0; i < jum; i++)
    {
        printf("Bilangan ke %d : ", i + 1);
        scanf("%d", &A[i]);
        cout << endl;
    }
    cout << endl;
    cout << "Hasil Pengurutan Secara Ascending : \n";
    cout << endl;
}

void buble(int jum)
{
    int i, j, temp;
    for(i = 1; i <= jum - 1; i++)
    {
        for(j = i; j < n; j++)
        {
            if(A[i - 1] > A[j])
            {
                temp = A[i - 1];
                A[i - 1] = A[j];
                A[j] = temp;
            }
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
}
    }
}

void output(int jum)
{
    int i;
    for(i = 0; i < jum; i++)
    {
        printf("Bilangan ke %d = %d\n", i + 1, A[i]);
        cout << endl;
    }
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

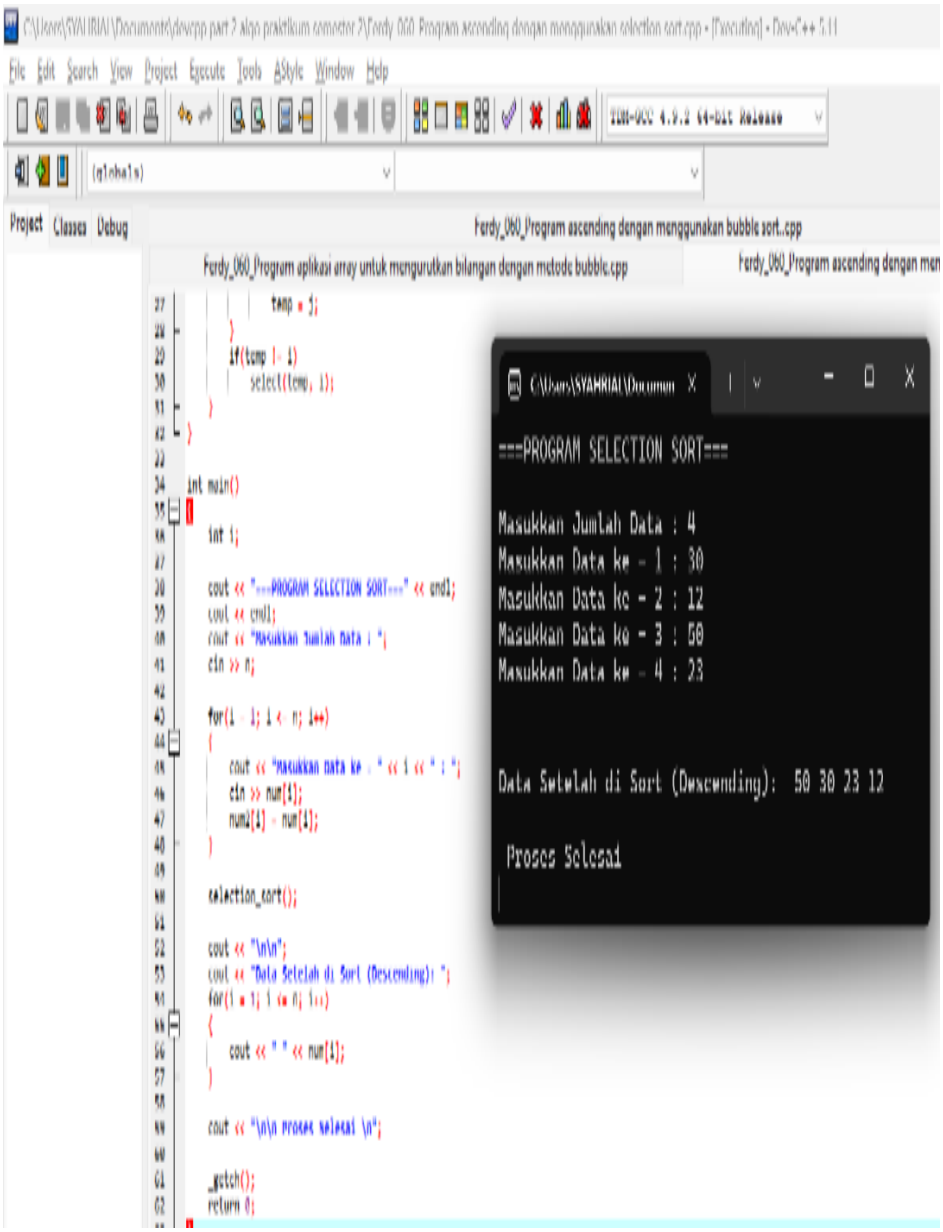
3. Program *ascending* dengan menggunakan *selection sort.*

```
#include <iostream>
#include <conio.h>

using namespace std;

int num[10], num2[10];
int n;

void select(int a, int b)
{
    int t;
    t = num[b];
    num[b] = num[a];
    num[a] = t;
}

void selection_sort()
{
    int temp, i, j;
    for(i = n; i > 0; i--)
    {
        temp = i;
        for(j = 1; j <= i; j++)
        {
            if(num[j] < num[temp])
                temp = j;
        }
        if(temp != i)
            select(temp, i);
    }
}

int main()
{
    int i;

    cout << "===PROGRAM SELECTION SORT===" << endl;
    cout << endl;
    cout << "Masukkan Jumlah Data : ";
    cin >> n;

    for(i = 1; i <= n; i++)
    {
        cout << "Masukkan Data ke - " << i << " : ";
        cin >> num[i];
        num2[i] = num[i];
    }

    selection_sort();

    cout << "\n\n";
    cout << "Data Setelah di Sort (Descending): ";
    for(i = 1; i <= n; i++)
    {
        cout << " " << num[i];
    }
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
    cout << "\n\n Proses Selesai \n";

    _getch();
    return 0;
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

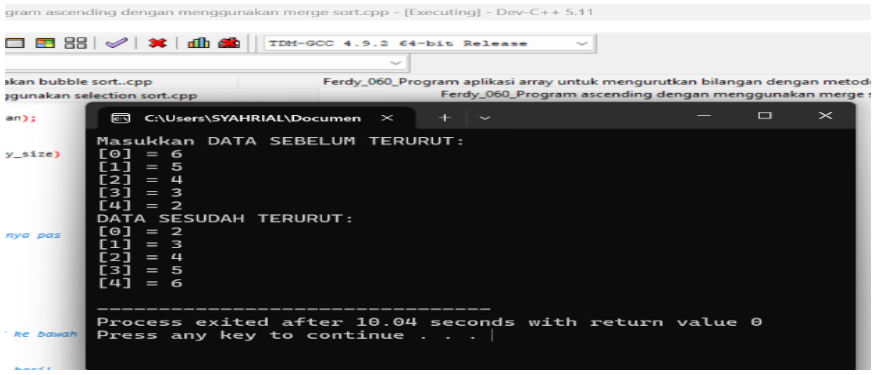
4. Program *ascending* dengan menggunakan *merge sort.*

```
#include <stdio.h>
#include <conio.h>

#define MAX 5

int data[MAX];
int temp[MAX];

void merge(int data[], int temp[], int kiri, int tengah,
int kanan)
{
    int i, left_end, num_elements, tmp_pos;
    left_end = tengah - 1;
    tmp_pos = kiri;
    num_elements = kanan - kiri + 1;

    while((kiri <= left_end) && (tengah <= kanan))
    {
        if(data[kiri] <= data[tengah])
        {
            temp[tmp_pos] = data[kiri];
            tmp_pos = tmp_pos + 1;
            kiri = kiri + 1;
        }
        else
        {
            temp[tmp_pos] = data[tengah];
            tmp_pos = tmp_pos + 1;
            tengah = tengah + 1;
        }
    }

    while(kiri <= left_end)
    {
        temp[tmp_pos] = data[kiri];
        kiri = kiri + 1;
        tmp_pos = tmp_pos + 1;
    }

    while(tengah <= kanan)
    {
        temp[tmp_pos] = data[tengah];
        tengah = tengah + 1;
        tmp_pos = tmp_pos + 1;
    }

    for(i = 0; i < num_elements; i++)
    {
        data[kanan] = temp[kanan];
        kanan = kanan - 1;
    }
}

void m_sort(int data[], int temp[], int kiri, int kanan)
{
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
int tengah;
    if(kanan > kiri)
    {
        tengah = (kanan + kiri) / 2;
        m_sort(data, temp, kiri, tengah);
        m_sort(data, temp, tengah + 1, kanan);
        merge(data, temp, kiri, tengah + 1, kanan);
    }
}

void mergesort(int data[], int temp[], int array_size)
{
    m_sort(data, temp, 0, array_size - 1);
}

int main()
{
    int i;

    // Disesuaikan agar huruf kapital dan spasinya pas
    printf("Masukkan DATA SEBELUM TERURUT:\n");

    for(i = 0; i < MAX; i++){
        // Mengubah format input menjadi [0] =
        printf("[%d] = ", i);
        scanf("%d", &data[i]);
    }

    mergesort(data, temp, MAX);

    // Disesuaikan menjadi DATA SESUDAH TERURUT ke bawah
    printf("DATA SESUDAH TERURUT:\n");
    for(i = 0; i < MAX; i++)
    {
        // Mengubah format output menjadi [0] = hasil
        printf("[%d] = %d\n", i, data[i]);
    }

    _getch();
    return 0;
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tugas !

1. Buatlah program untuk mengurutkan data-data berikut ini secara *Descending* dengan *Bubble Sort.*

   4, 8, 5, 9, 6, 2, 7, 5, 9, 5, 23, 30

2. Buatlah program untuk mengurutkan data-data berikut ini secara *Descending* dengana metode *Selection Sort.*

3. Buatlah program untuk mengurutkan data-data berikut ini secara *Descending* dengan metode *Merger Sort.*

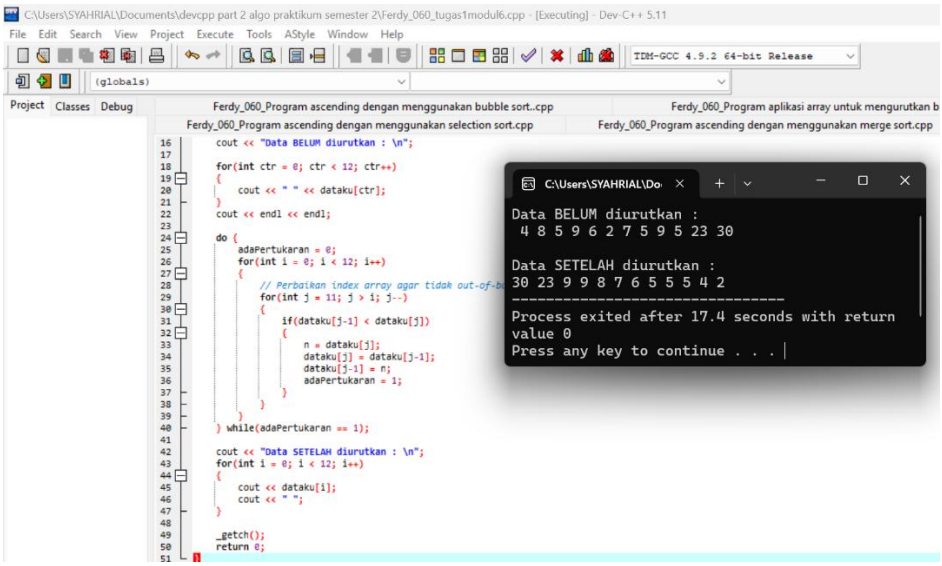
1. Buatlah program untuk mengurutkan data – data berikut ini secara descending dengan bubble sort (4, 8, 5, 9, 6, 2, 7, 5, 9, 5, 23, 30)

```
#include <iostream>
#include <conio.h>
#include <iomanip>

using namespace std;

int main()
{
    int dataku[] = {4, 8, 5, 9, 6, 2, 7, 5, 9, 5, 23,
30};
    int adaPertukaran;
    int n;

    cout << "Data BELUM diurutkan : \n";

    for(int ctr = 0; ctr < 12; ctr++)
    {
        cout << " " << dataku[ctr];
    }
    cout << endl << endl;

    do {
        adaPertukaran = 0;
        for(int i = 0; i < 12; i++)
        {
            // Perbaikan index array agar tidak out-of-
bounds (dimulai dari 11)
            for(int j = 11; j > i; j--)
            {
                if(dataku[j-1] < dataku[j])
                {
                    n = dataku[j];
                    dataku[j] = dataku[j-1];
                    dataku[j-1] = n;
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
                    adaPertukaran = 1;
                }
            }
        }
    } while(adaPertukaran == 1);

    cout << "Data SETELAH diurutkan : \n";
    for(int i = 0; i < 12; i++)
    {
        cout << dataku[i];
        cout << " ";
    }

    _getch();
    return 0;
}
```

Output :



Pada program di atas, kita menggunakan std::vector<int> untuk menyimpan data yang akan diurutkan. Kemudian, terdapat fungsi bubbleSortDescending yang menerima referensi ke vektor data untuk diurutkan secara descending.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

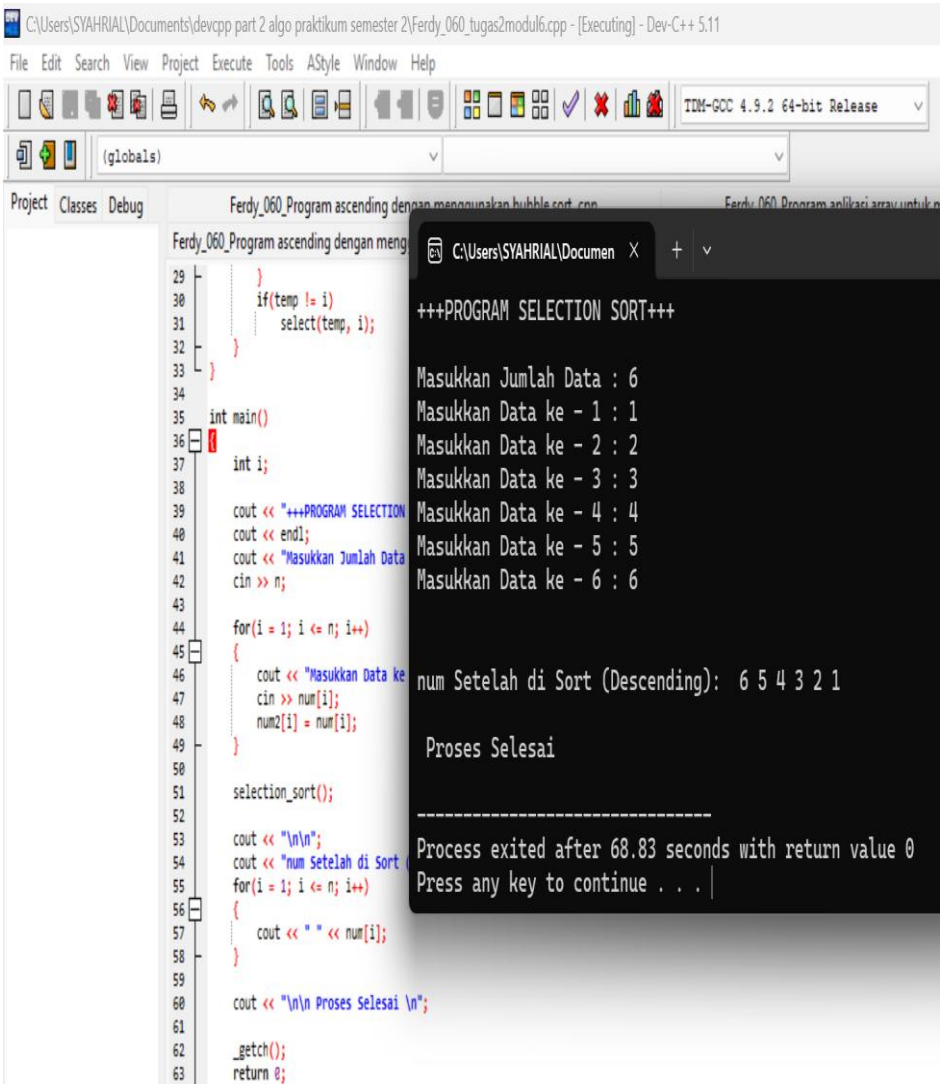
2. Jawablah program untuk mengurutkan data – data berikut ini secara descending dengan metode selection sort.

```
#include <iostream>
#include <conio.h>

using namespace std;

int num[10], num2[10];
int n;

void select(int a, int b)
{
    int t;
    t = num[b];
    num[b] = num[a];
    num[a] = t;
}

void selection_sort()
{
    int temp, i, j;
    for(i = n; i > 0; i--)
    {
        temp = i;
        for(j = 1; j <= i; j++)
        {
            if(num[j] < num[temp])
                temp = j;
        }
        if(temp != i)
            select(temp, i);
    }
}

int main()
{
    int i;

    cout << "+++PROGRAM SELECTION SORT+++" << endl;
    cout << endl;
    cout << "Masukkan Jumlah Data : ";
    cin >> n;

    for(i = 1; i <= n; i++)
    {
        cout << "Masukkan Data ke - " << i << " : ";
        cin >> num[i];
        num2[i] = num[i];
    }

    selection_sort();

    cout << "\n\n";
    cout << "num Setelah di Sort (Descending): ";
    for(i = 1; i <= n; i++)
    {
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
cout << " " << num[i];
    }

    cout << "\n\n Proses Selesai \n";

    _getch();
    return 0;
}
```

Output :



Dengan menjalankan program ini, Anda dapat mengurutkan data secara descending menggunakan metode Selection Sort. Program akan meminta jumlah data yang akan diurutkan, kemudian meminta data tersebut, dan akhirnya menampilkan data setelah diurutkan secara descending.

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

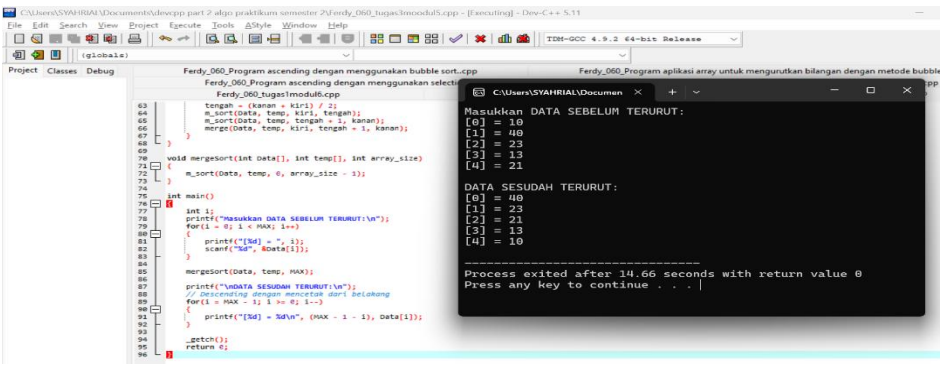
3. Buatlah program untuk mengurutkan data – data berikut ini secara descending dengan metode merge sort.

```
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;
#define MAX 5

int Data[MAX];
int temp[MAX];

void merge(int Data[], int temp[], int kiri, int tengah,
int kanan)
{
    int i, left_end, num_elements, tmp_pos;
    left_end = tengah - 1;
    tmp_pos = kiri;
    num_elements = kanan - kiri + 1;

    while((kiri <= left_end) && (tengah <= kanan))
    {
        if(Data[kiri] <= Data[tengah])
        {
            temp[tmp_pos] = Data[kiri];
            tmp_pos = tmp_pos + 1;
            kiri = kiri + 1;
        }
        else
        {
            temp[tmp_pos] = Data[tengah];
            tmp_pos = tmp_pos + 1;
            tengah = tengah + 1;
        }
    }

    while(kiri <= left_end)
    {
        temp[tmp_pos] = Data[kiri];
        kiri = kiri + 1;
        tmp_pos = tmp_pos + 1;
    }

    while(tengah <= kanan)
    {
        temp[tmp_pos] = Data[tengah];
        tengah = tengah + 1;
        tmp_pos = tmp_pos + 1;
    }

    // PERBAIKAN: Menggunakan '<' agar tidak out of bounds
    for(i = 0; i < num_elements; i++)
    {
        Data[kanan] = temp[kanan];
        kanan = kanan - 1;
    }
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
}

void m_sort(int Data[], int temp[], int kiri, int kanan)
{
    int tengah;
    if(kanan > kiri)
    {
        tengah = (kanan + kiri) / 2;
        m_sort(Data, temp, kiri, tengah);
        m_sort(Data, temp, tengah + 1, kanan);
        merge(Data, temp, kiri, tengah + 1, kanan);
    }
}

void mergeSort(int Data[], int temp[], int array_size)
{
    m_sort(Data, temp, 0, array_size - 1);
}

int main()
{
    int i;
    printf("Masukkan DATA SEBELUM TERURUT:\n");
    for(i = 0; i < MAX; i++)
    {
        printf("[%d] = ", i);
        scanf("%d", &Data[i]);
    }

    mergeSort(Data, temp, MAX);

    printf("\nDATA SESUDAH TERURUT:\n");
    // Descending dengan mencetak dari belakang
    for(i = MAX - 1; i >= 0; i--)
    {
        printf("[%d] = %d\n", (MAX - 1 - i), Data[i]);
    }

    _getch();
    return 0;
}
```

Output :



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

DAFTAR PUSTAKA

Uce Indahyanti, Buku Ajar Algoritma Pemrograman Dalam Bahasa C++, UMSIDA Press, 2020

Abdul Kadir, Logika Pemrograman Python, Elex Media Komputindo, 2019

Rinaldi Munir, Algoritma dan Pemrograman Dalam Bahasa Pascal , C dan C++, Edisi Keenam. Bandung : Informatika, 2016.

Modul Laboratorium Algoritma dan Struktur Data, Fakultas Teknik - Prodi Informatika - UMSIDA, 2017

Cipta Ramadhani, Algoritma Pemrograman Dan Struktur Data Menggunakan C++, Yogyakarta : ANDI, 2019

NL Azizah.U Indahyanti. Anonymously Tracking Covid-19 Patient Using Labeling Graph Approach And GPS Tracking Technology. Proceeding ICCGANT 2021

QS Alaq 3-5 (pentingnya iptek) dan QS Ali Imron 190-191 (berpikir kritis)

Fachrurrozi M., Modul Praktikum Struktur Data, Laboratorium Universitas Brawijaya Malang, Malang, 2006.

Haryanto Bambang, Struktur Data, Informatika, Bandung, 2008.

Kristanto Andi, Struktur Data dengan C++, Graha Ilmu, Yogyakarta, 2009.

Warni Elly, Buku Bahan Ajar Struktur Data, Universitas Hasanuddin, 2012.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

KARTU ASISTENSI
PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 – 2026

Nama : Mochammad Ferdyansyah
NIM : 251080200060
Kelompok : 6

No	Judul Praktikum	Tanggal Praktikum	Tanggal Asistensi	Catatan Asisten
1.	Struktur Data, Array, Pointer, Dan Struktur	05 Mei 2026	13/5 2026	Baik
2.	Linked List (Senarai)	05 Mei 2026	13/5 2026	Baik
3.	Stack (Tumpukan)	05 Mei 2026	13/5 2026	Baik
4.	Queue (Antrian)	30 Juni 2026	6/7 2026	Sangat Baik
5.	Rekursif	30 Juni 2026	6/7 2026	sangat Baik
6.	Sorting (Pengurutan)	30 Juni 2026	6/7 2026	Sangat Baik

Mengetahui,

Dosen Praktikum

(Yunianita Rahmawati, S.Kom. M.Kom)

Asisten Praktikum 6/7 2026

(Nadya Mulya Islami)