

Tim Pengajar IF2250

IF2250 – Rekayasa Perangkat Lunak Pengantar Kuliah

SEMESTER II - 2023/2024



KNOWLEDGE & SOFTWARE ENGINEERING

Deskripsi

- IF2250 Rekayasa Perangkat Lunak
- Kuliah Wajib - 3 SKS
- Jadwal:
 - Senin: 13.00-14.40
 - Jumat: 15.00-15.50
- Dosen:
 - K1: Yani Widayani (yani@informatika.org)
 - K2: Wikan Danar Sunindyo (wikan@informatika.org)
 - K3: Yani Widayani/M.R. Alghazali
(yani@informatika.org / algo@informatika.org)

Tujuan Umum

- Memberikan pengetahuan tentang rekayasa perangkat lunak secara lengkap dan kemampuan merekayasa perangkat lunak skala kecil sampai menengah

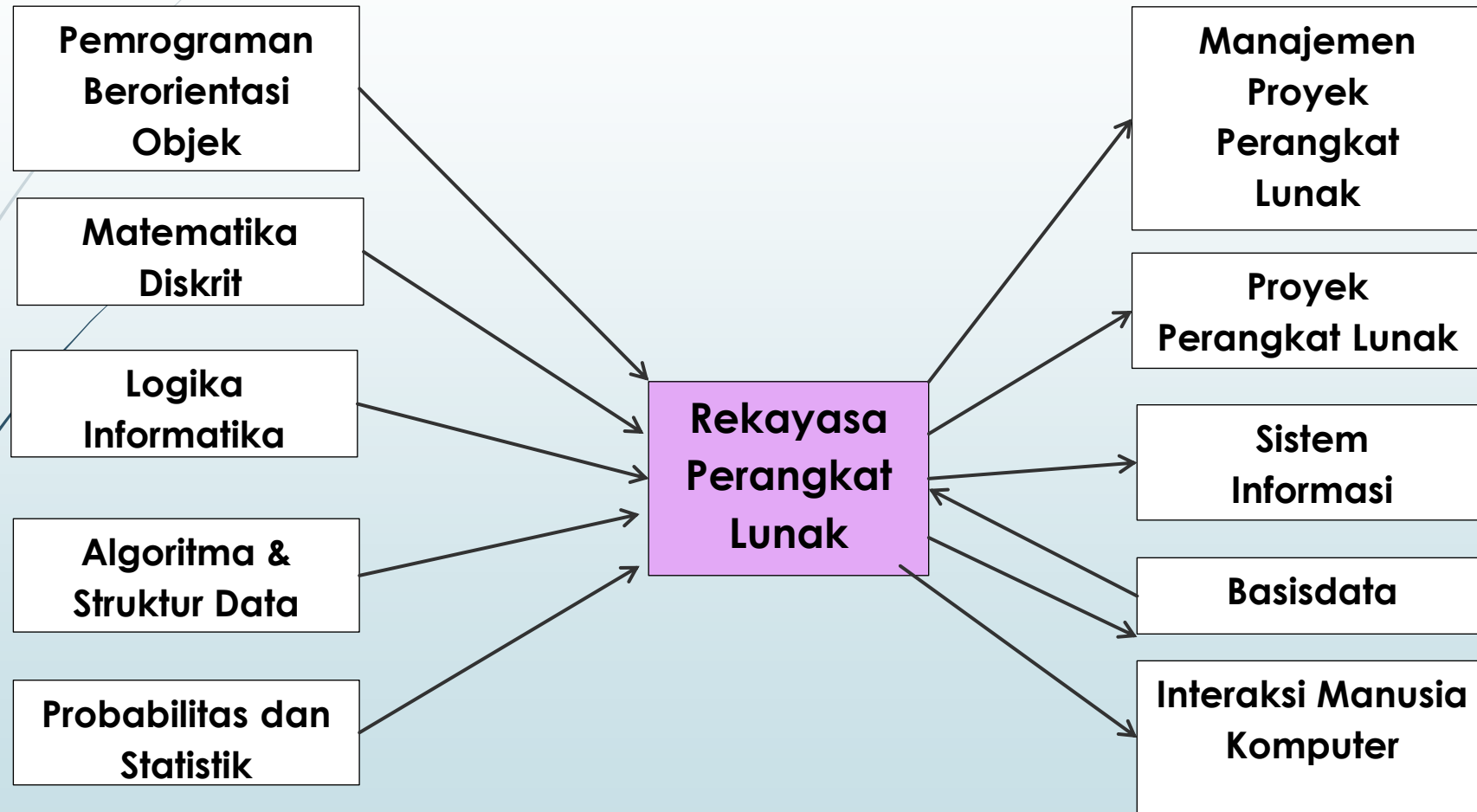
Tujuan Khusus

- Peserta diharapkan:
 - Memahami **pengertian** perangkat lunak dan rekayasa perangkat lunak
 - Memahami **posisi** rekayasa perangkat lunak dalam rekayasa sistem
 - Memahami berbagai **model proses** pembangunan perangkat lunak; termasuk di dalamnya siklus hidup perangkat lunak; serta dapat memilih model proses yang tepat untuk kondisi tertentu
 - Memahami **requirement** (kebutuhan) sebuah perangkat lunak ; dan dapat memodelkannya dalam beberapa pendekatan
 - Memahami **konsep** perancangan dan dapat menerapkannya dalam perancangan perangkat lunak skala kecil hingga menengah

Tujuan Khusus

- Peserta diharapkan:
 - Memahami **perancangan arsitektural** dan dapat menerapkannya dalam perancangan perangkat lunak skala kecil hingga menengah
 - Memahami **perancangan level komponen** perangkat lunak dan dapat menerapkannya dalam perancangan perangkat lunak skala kecil hingga menengah
 - Memahami **perancangan antarmuka** dan dapat menerapkannya dalam perancangan perangkat lunak skala kecil hingga menengah
 - Mengetahui beberapa **faktor kualitas** perangkat lunak dan memahami pentingnya kualitas perangkat lunak
 - Mengetahui beberapa **teknik pengujian** perangkat lunak

Peta Materi RPL



**T
U
G
A
S

A
K
H
I
R**

Perangkat Lunak?

1. **Kumpulan Instruksi** (program komputer) yang ketika dieksekusi menyediakan fitur, fungsi, dan kinerja yang diinginkan
2. **Struktur Data**, yang memungkinkan program memanipulasi informasi
3. **Dokumentasi**, yang menggambarkan operasi dan penggunaan program

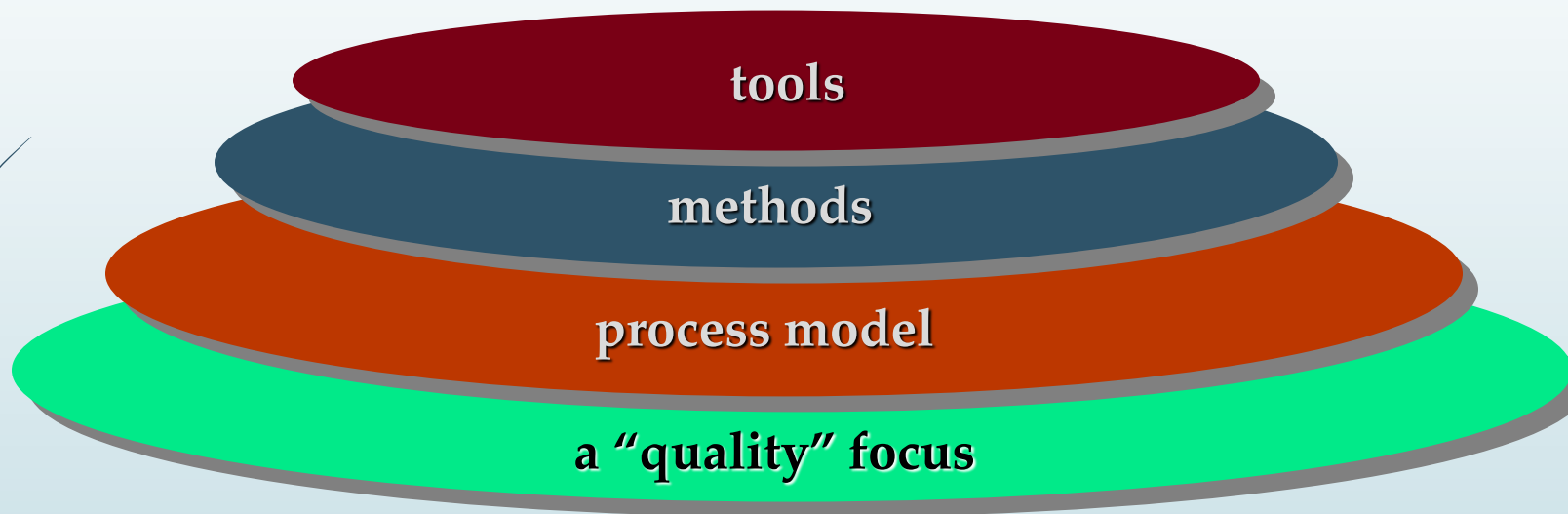
Rekayasa Perangkat Lunak?

- *Software engineer* kerap kali menemukan permasalahan terkait komputer atau sistem komputer yang ada, namun terkadang permasalahan yang mendasarinya tidak ada hubungannya dengan komputer.
- Oleh karena itu, kita perlu memahami dahulu sifat permasalahannya:
 - Jangan memaksakan mesin atau teknik komputasi pada semua masalah.
 - Cari solusinya lebih dahulu dan jika diperlukan maka digunakan teknologi sebagai alat untuk mengimplementasikan solusi kita

Rekayasa Perangkat Lunak?

- Dalam memecahkan masalah, kita gunakan berbagai **metode, tools, prosedur, dan paradigma**.
 - **Metode**: prosedur (formal) yang ditempuh untuk mencapai tujuan (memproduksi hasil) tertentu
 - **Teknik**: cara yang spesifik dalam pemecahan masalah tertentu yang ditemukan dalam pelaksanaan prosedur
 - **Tool**: sebuah instrumen atau sistem otomatisasi untuk mencapai sesuatu dengan lebih baik
 - **Prosedur**: kombinasi tools dan teknik yang menghasilkan produk tertentu
 - **Paradigma**: pendekatan atau filosofi tertentu untuk membangun perangkat lunak.

A Layered Technology



Pengelolaan Jadwal Penerbangan Domestik

(Versi: 14 April 2014)

DESKRIPSI PERSOALAN

Sebuah instansi koordinator bandar udara nasional mengelola seluruh jadwal penerbangan domestik di Indonesia. Sebuah rencana dan implementasi penerbangan direpresentasikan oleh nomor penerbangan (*flight_no*, misalnya: GA101), kode bandara/kota asal (misalnya Bandung: BDO), kode bandara/kota tujuan (misalnya Surabaya, SUB), rencana tanggal dan jam keberangkatan (misalnya, 1 April 2014, 10.00), rencana tanggal dan jam kedatangan (misalnya, 1 April 2014, 11.15), implementasi tanggal dan jam keberangkatan (misalnya, 1 April 2014, 11.00), implementasi tanggal dan jam kedatangan (misalnya, 1 April 2014, 12.30).

Sebuah kode penerbangan dimungkinkan untuk dipakai dalam sebuah penerbangan *direct flight* maupun *non direct flight*. Pada *non direct flight*, dimungkinkan adanya penerbangan yang berturutan di mana ada satu atau lebih kota/bandara antara kota/bandara asal ke kota/bandara tujuan akhir. Contoh, sebuah penerbangan GA121 dari Bandung (CGK) ke Denpasar (DPS) melalui Surabaya (SUB), direpresentasikan dengan 2 penerbangan berurutan, yaitu GA121 BDO-SUB, dan GA121 SUB-DPS

Sebuah program digunakan untuk membantu instansi tersebut untuk mengelola jadwal penerbangan yang dimaksud. Jadwal penerbangan dapat dibaca oleh program dari sebuah file teks dengan setiap barisnya menggambarkan sebuah penerbangan dengan format sbb:

```
FlightNo | KotaAsal | KotaTujuan | RencanaTanggalBerangkat(dd:mm:yyyy) |
RencanaJamBerangkat(hh:mm) | RencanaTanggalKedatangan | RencanaJanKedatangan |
ImplementasiTanggalBerangkat | ImplementasiJamBerangkat | ImplementasiTanggalKedatangan |
ImplementasiJanKedatangan
```

Contoh:

```
GA121 | BDO | SUB | 01:04:2014 | 10:00 | 01:04:2014 | 11:15 | 01:04:2014 | 11:00 |
01:04:2014 | 12:30
```

IF2250-RPL

Sebuah rencana penerbangan dalam file jadwal penerbangan belum tentu memiliki implementasi penerbangan, sehingga sebuah baris dalam file tsb dapat berisi informasi misalnya sbb

```
GA121 | BDO | SUB | 01:04:2014 | 10:00 | 01:04:2014 | 11:15 | __:__:__ | __:__ |
__:__:__ | __:__
```

Perhatikan bahwa:

1. meskipun informasi implementasi waktu keberangkatan dan waktu kedatangan belum terdefiniskan, pembatas “|” tetap dituliskan dalam file, dan tanda *underscore* (“_”) digunakan sebagai pengganti digit data yang belum ada/belum tersedia
2. setiap field memiliki panjang yang tetap. Sebagai contoh, kode penerbangan memiliki panjang 5 karakter, sedangkan kode bandara/kota keberangkatan/kedatangan memiliki panjang 3 karakter.
3. Dalam satu file, nomor penerbangan belum tentu unik (karena *non direct flight*), tetapi gabungan antara nomor penerbangan, bandara/kota asal, bandara/kota tujuan selalu unik.

TUGAS

Buatlah sebuah program dengan bahasa PASCAL yang mengelola jadwal penerbangan tersebut dengan primitif-primitif program sbb:

1. **P1-load:** Membaca data penerbangan dari file
2. **P2-save:** Menyimpan data penerbangan ke dalam file
3. **P3-add:** Menambah data rencana penerbangan
4. **P4-mod:** Mengubah data rencana penerbangan
5. **P5-display:** Menampilkan seluruh jadwal penerbangan yang sedang dikelola (baik data dari file maupun data yang ditambahkan selama program dieksekusi)
6. **P6-modimpl:** Mengubah data implementasi penerbangan
7. **P7-displaydate:** Menampilkan jadwal penerbangan (*direct flight*) pada tanggal keberangkatan tertentu, terurut waktu keberangkatan
8. **P8-displaydirect:** Menampilkan jadwal penerbangan (*direct flight*) pada keberangkatan tertentu, dari kota/bandara tertentu menuju kota/bandara tertentu, terurut waktu keberangkatan
9. **P9-displayintime:** Menampilkan penerbangan-penerbangan yang tepat waktu (rencana dan implementasi sama)
10. **P10-displaylate:** Menampilkan penerbangan yang interval waktu penerbangannya lebih lama dari yang direncanakan
11. **P11-displayearlier:** Menampilkan penerbangan yang interval waktu penerbangannya lebih cepat dari yang direncanakan
12. **P12-searchflight:** Menemukan (search) dan menampilkan data penerbangan, berdasarkan masukan pengguna, yaitu kode penerbangan, asal keberangkatan, dan tanggal keberangkatan
13. **P13-searchlatest:** Menemukan (search) dan menampilkan data penerbangan terakhir dari sebuah bandara asal ke sebuah bandara tujuan pada tanggal tertentu
14. **P14-sort:** Menampilkan seluruh jadwal penerbangan dari satu bandara ke bandara tertentu, terurut menaik berdasarkan tanggal dan jam keberangkatan.
15. **P15-exit:** Keluar dari program (exit)

Bonus Implementasi Primitif

Primitif tambahan berikut tidak harus diimplementasikan dalam program Anda, tetapi apabila diimplementasikan akan menambah nilai tugas anda maksimum **10% dari skala penilaian normal**:

1. **B1-displaynondirect:** Program dapat memfasilitasi pencarian jadwal penerbangan *non direct flight*
2. **B2-displaytransit:** Program dapat memfasilitasi pencarian jadwal penerbangan dengan transit (kode penerbangan berbeda, tetapi dengan jadwal berturutan dan bandara yang tersambung). Contoh: penerbangan: GA121, BDG-SUB, 10 April 2014, 10.00, 10 April 2014, 11.15 yang dilanjutkan dengan JT331, SUB-DPS, 10 April 2014, 12.30, 10 April 2014, 13.30
Catatan: untuk dapat dihubungkan sebagai penerbangan dengan transit, selisih kedatangan antara kedatangan penerbangan sebelumnya dengan keberangkatan penerbangan berikutnya berada dalam rentang 1-6 jam.
3. **B3-unit:** Program dibuat dengan memanfaatkan unit dalam Pascal. Struktur modul/unit yang dipakai harus dideskripsikan dalam laporan.

ANTAR MUKA DAN CARA MEMBERIKAN COMMAND

Program dibuat dengan antarmuka berbasis teks. Program menyediakan prompt kepada user dalam bentuk karakter ">", dan user memberikan perintah (*command*) kepada program dalam bentuk teks perintah tertentu. Berikut adalah beberapa contoh perintah dan hasil eksekusi dari program. Perhatikan bahwa teks yang dengan karakter tebal (**bold**) adalah teks perintah atau input yang dimasukkan oleh user. Sebaiknya, teks yang tidak bercetak tebal adalah teks yang merupakan output oleh program, baik berupa teks instruksi bagi user untuk memberikan input, pesan kesalahan, maupun data yang ditampilkan program.

1. **P1-load**: Membaca data penerbangan dari file

```
> load
> nama file: C:\flightschedule1.txt
> pembacaan file berhasil
>
> load
> nama file: C:\filetidakada.txt
> pembacaan file tidak berhasil, file tidak ditemukan
>
```

2. **P2-save**: Menyimpan data penerbangan ke dalam file

```
> save
> nama file: c:\flightschedule2.txt
> penyimpanan file berhasil
>
> save
> nama file: c:\filesalah#@.txt
> penyimpanan file tidak berhasil
>
```

3. **P3-add**: Menambah data rencana penerbangan

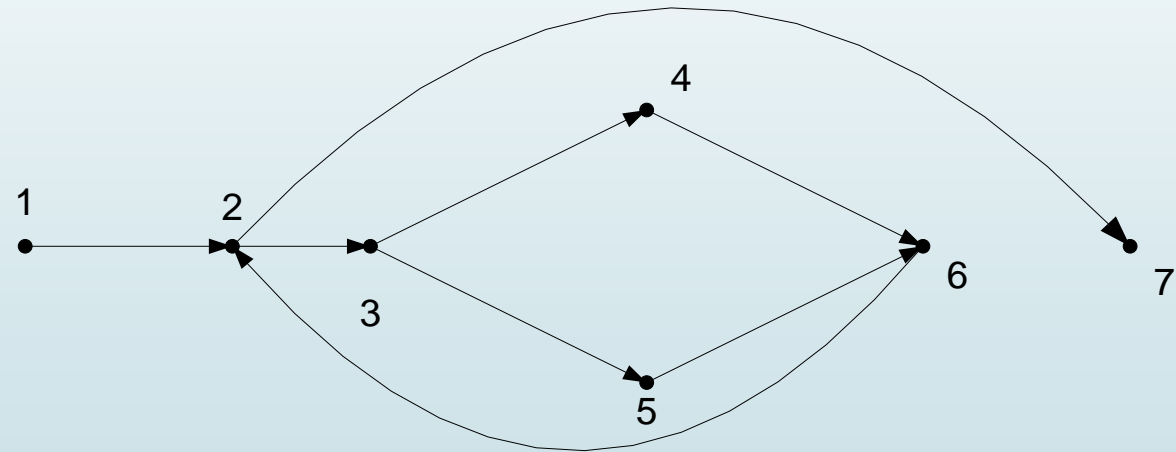
```
> add
> kode penerbangan: GA313
> kode bandara keberangkatan: BDO
> kode bandara kedatangan: SUB
> tanggal keberangkatan: 1
> bulan keberangkatan: 4
> tahun keberangkatan: 2014
> jam keberangkatan: 10
> menit keberangkatan: 30
> tanggal kedatangan: 1
> bulan kedatangan: 4
> tahun kedatangan: 2014
> jam kedatangan: 11
> menit kedatangan: 30
> penambahan jadwal rencana penerbangan berhasil
>
```

Graf

```

read(x);
while x <> 9999 do
begin
  if x < 0 then
    writeln('Masukan tidak boleh negatif')
  else
    x:=x+10;
  read(x);
end;
writeln(x);

```

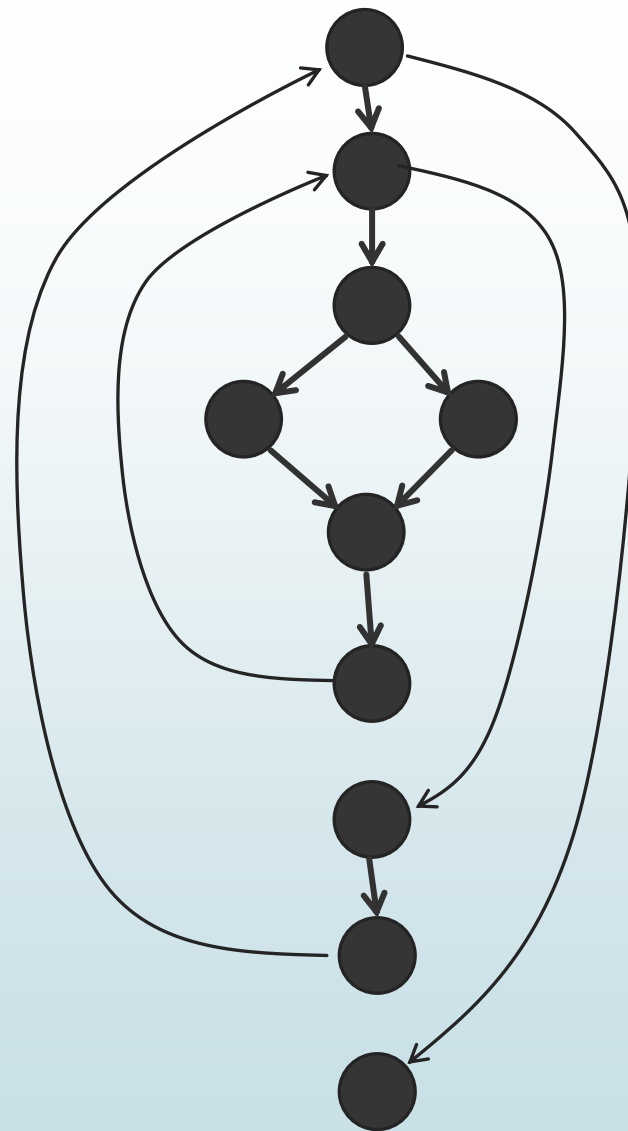
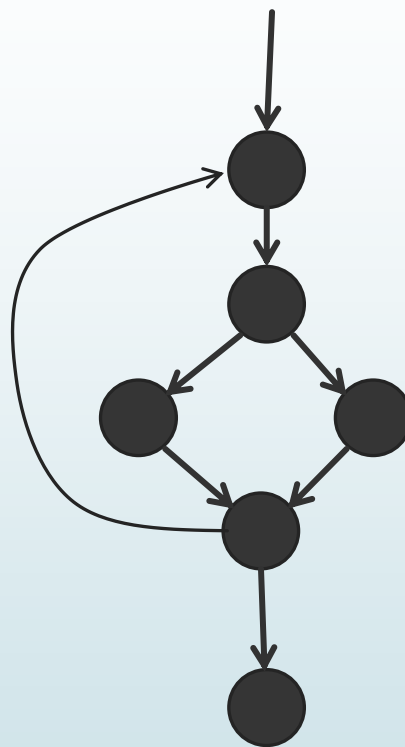
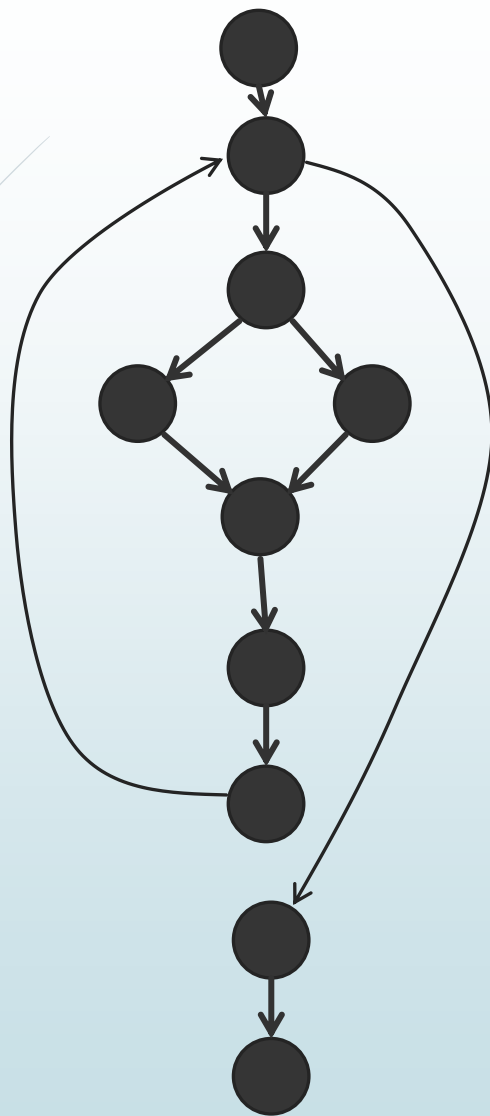


Keterangan:

1 : read(x)
 2 : x <> 9999
 3 : x < 0

4 : writeln('Masukan tidak boleh negatif');

5 : x := x + 10
 6 : read(x)
 7 : writeln(x)



Teori Himpunan?

Permutasi? Kombinasi?

Proposisi?

Induction Reasoning?

Deduction Reasoning?

Batasan-batasan.... (Constraints...)



Saya mau kamu buat program untuk membantu orang cari kendaraan dari suatu tempat ke tempat lain dengan mudah! !

Program harus **mudah dipakai**, harus **mudah dijalankan**, **tidak membosankan**, **banyak yang suka**, sehingga **berani membayar mahal**, sehingga **perusahaan untung**, sehingga tahun depan kita bisa **menambah karyawan baru**.



Biaya tidak boleh lebih dari 10 juta rupiah, waktu kamu 3 bulan!

Oya program harus bisa jalan di komputer, di handphone, juga di tablet ya...

Lingkup Bahasan

- Pendahuluan: Software dan Software Engineering; Software vs System
- Model Proses dan Agile Development
- Kebutuhan P/L (Requirement)
- Analisis:
 - Pemodelan dengan pendekatan terstruktur (Structured Approach)
 - Pemodelan dengan pendekatan berbasis Objek (Object-oriented Approach)
- Perancangan:
 - Konsep Perancangan
 - Perancangan Arsitektural
 - Perancangan Level Komponen
 - Perancangan Antarmuka
 - Perancangan dengan pendekatan terstruktur
 - Perancangan dengan pendekatan berbasis objek
- Pengujian P/L
- Software Development Tools

IF2250-RPL



Buku Pegangan

- Roger S. Pressman; Software Engineering: A Practitioner's Approach (8th Ed.); Mc Graw-Hill, 2015
- Ian Sommerville, Software Engineering (10th Ed.), Pearson, 2016

Komponen Penilaian

- Kuis (tidak terjadwal)
- Tugas
 - Analisis & Perancangan Terstruktur
 - Analisis & Perancangan Berorientasi Objek
 - Pembangunan Perangkat Lunak
 - Tugas Pengujian
 - PR atau tugas-tugas kecil
- Ujian:
 - UTS
 - UAS
- Kehadiran minimum 75% (untuk bisa mengikuti UAS)

Skenario Kuliah

- Mahasiswa dapat mendownload materi sebelum perkuliahan dimulai, dan wajib mempelajarinya terlebih dahulu;
- Tatap muka di kelas dapat digunakan untuk mereview materi yang telah dipelajari, latihan, dan diskusi;
- Pembahasan tugas dilakukan untuk kelompok tertentu yang akan ditentukan kemudian.

Tugas

- Tugas dilakukan per kelompok
 - 1 kelompok : 4-5 orang
- Jenis tugas
 - Tugas Analisis:
 - Terstruktur (Data flow diagram)
 - Objek (Use case diagram, Class diagram)
 - Tugas Perancangan
 - Perancangan Terstruktur
 - Perancangan Objek
 - Tugas Pembangunan Perangkat Lunak
 - Tugas Pengujian Fungsional

Yang harus dilakukan

- ❑ Isi presensi kehadiran di situs akademik.itb.ac.id;
- ❑ Pelajari materi di MsTeams;
 - Join Teams IF2250 RPL 2024 dg kode: **p55tqbj**
 - Sediakan sendiri referensi yang diperlukan;
- ❑ Cek media komunikasi:
 - memberikan pengumuman di:
Teams IF2250 Rekayasa Perangkat Lunak 2024