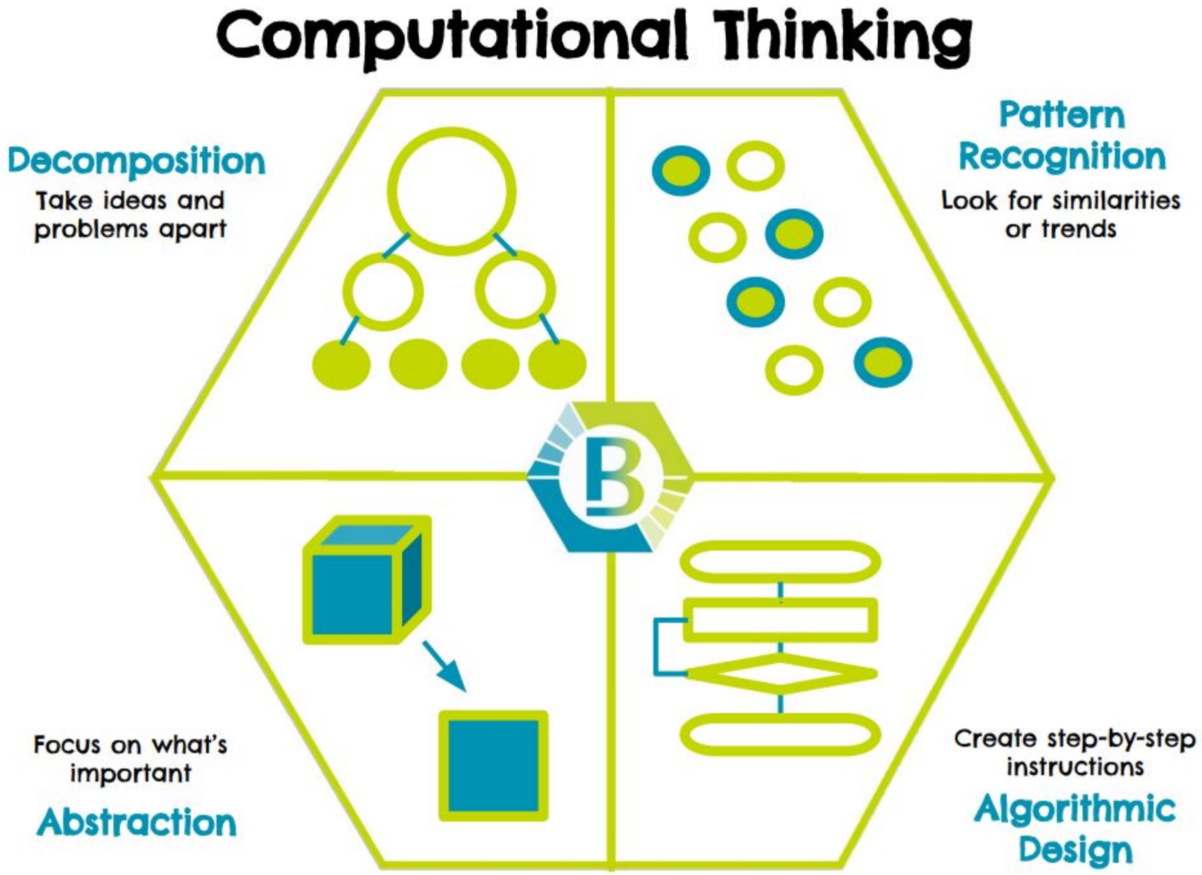
UNIVERSITAS TRISAKTI
PRODI TEKNIK PERTAMBANGAN

DECOMPOSITION

MFU6207 - KOMPUTASI DASAR (*COMPUTATIONAL THINKING*)

6th session
1.5 hours

Lecturer :
Danu Putra, S.T., M.T.



www.wcpss.net

SEMESTER SYLLABUS

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term

DEFINITION

Heuristic and Decomposition

A heuristic is **a specific strategy** in problem-solving that usually yields **a good enough answer**. though not necessarily an optimal one. Examples include trial and error, using a **rule of thumb or drawing an analogy**.

Computational thinking promotes one of these heuristics to a core practice: decomposition, which is an approach that seeks to **break a complex problem down into simpler parts** that are easier to deal with

Decomposition is a **divide-and-conquer strategy**, something seen in numerous places outside computing:

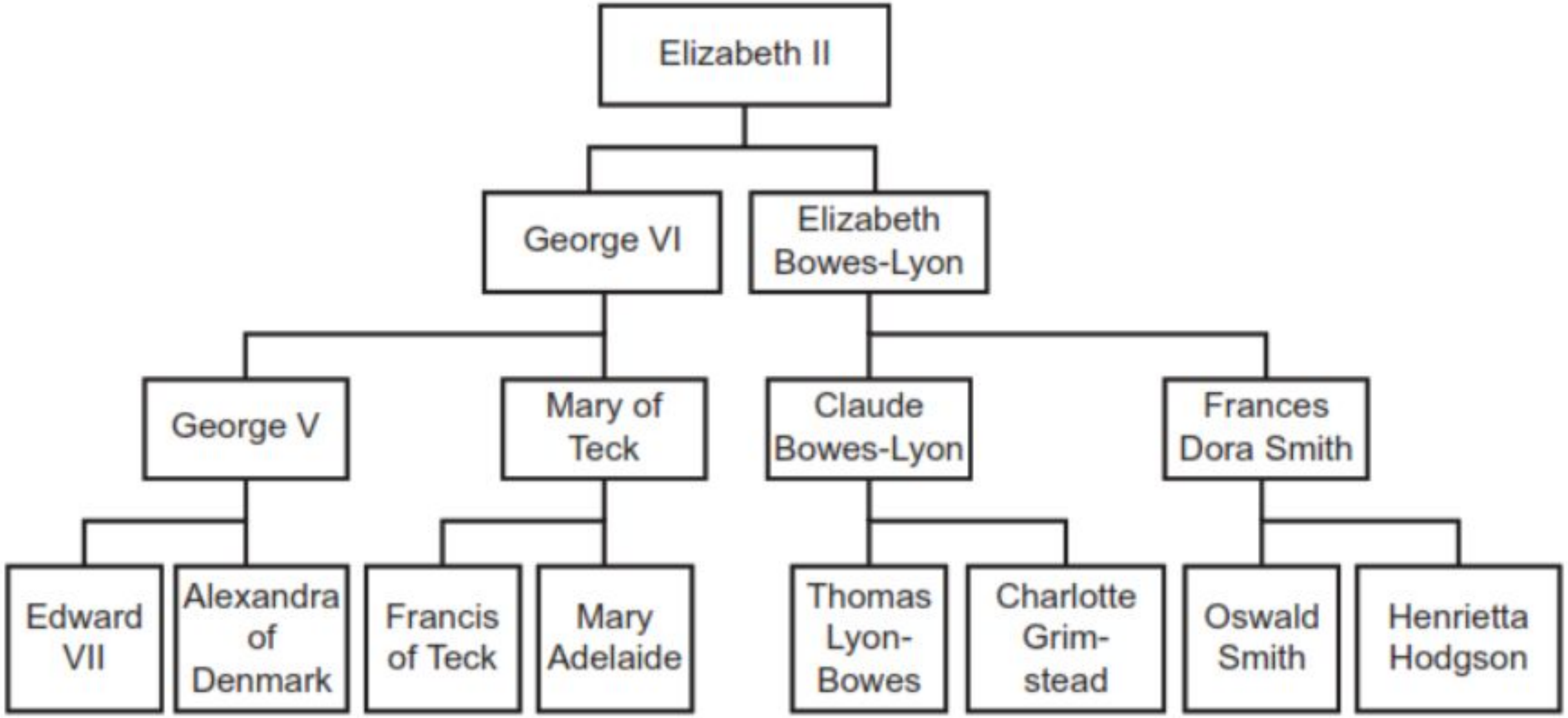
- Generals employ it on the battlefield when outnumbered by the enemy. By **engaging only part of the enemy forces**, they neutralise their opponent's advantage of numbers and **defeat them one group at a time**.
- Politicians use it to **break opposition up into weaker parties** who might otherwise unite into a stronger whole.
- When faced with a large, diverse audience, **marketers segment their potential customers** into different stereotypes and target each one differently.

Karl Breecher, 2017

DEFINITION

Recursion and Tree Structure

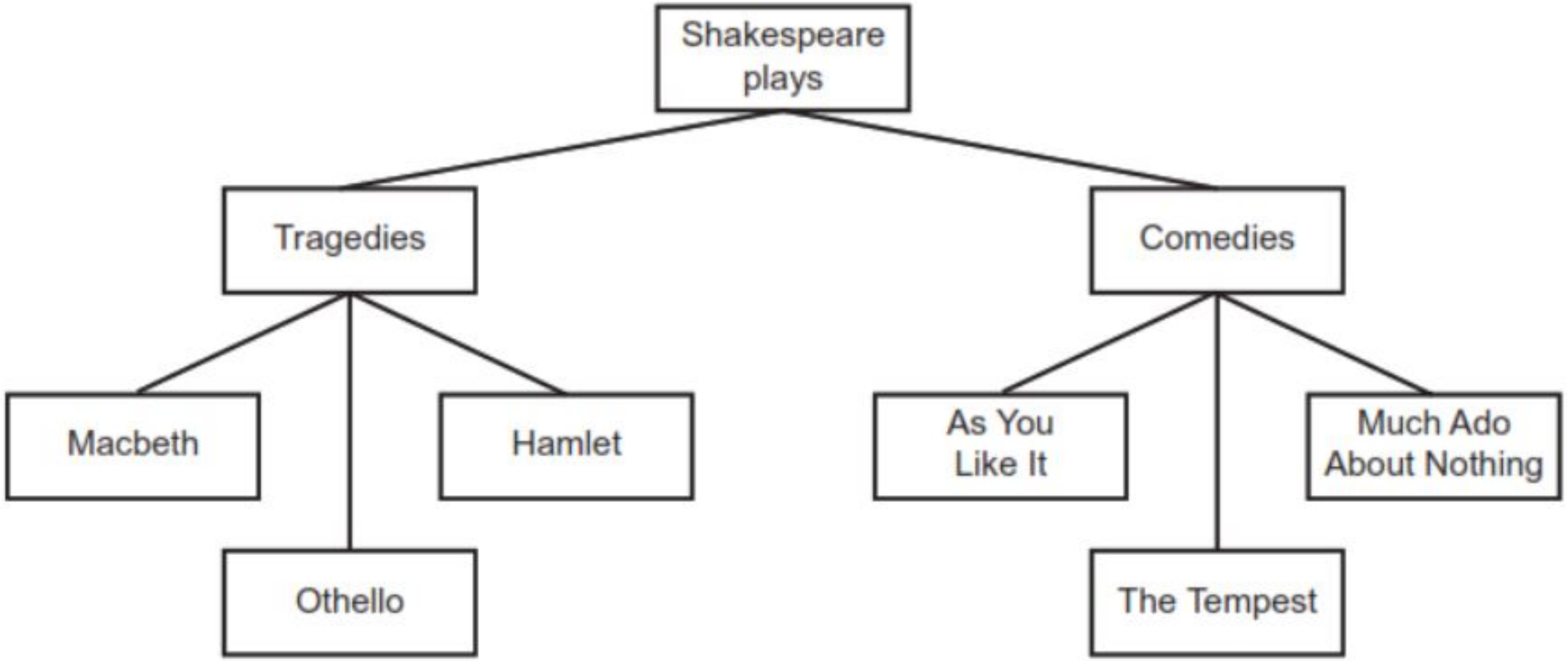
Recursion: **a technique** used to simplify a problem. It defines the solution to a large, complex problem in terms of smaller, simpler problems of the same form as the original problem.



Karl Breecher, 2017

EXAMPLES

Tree Structure



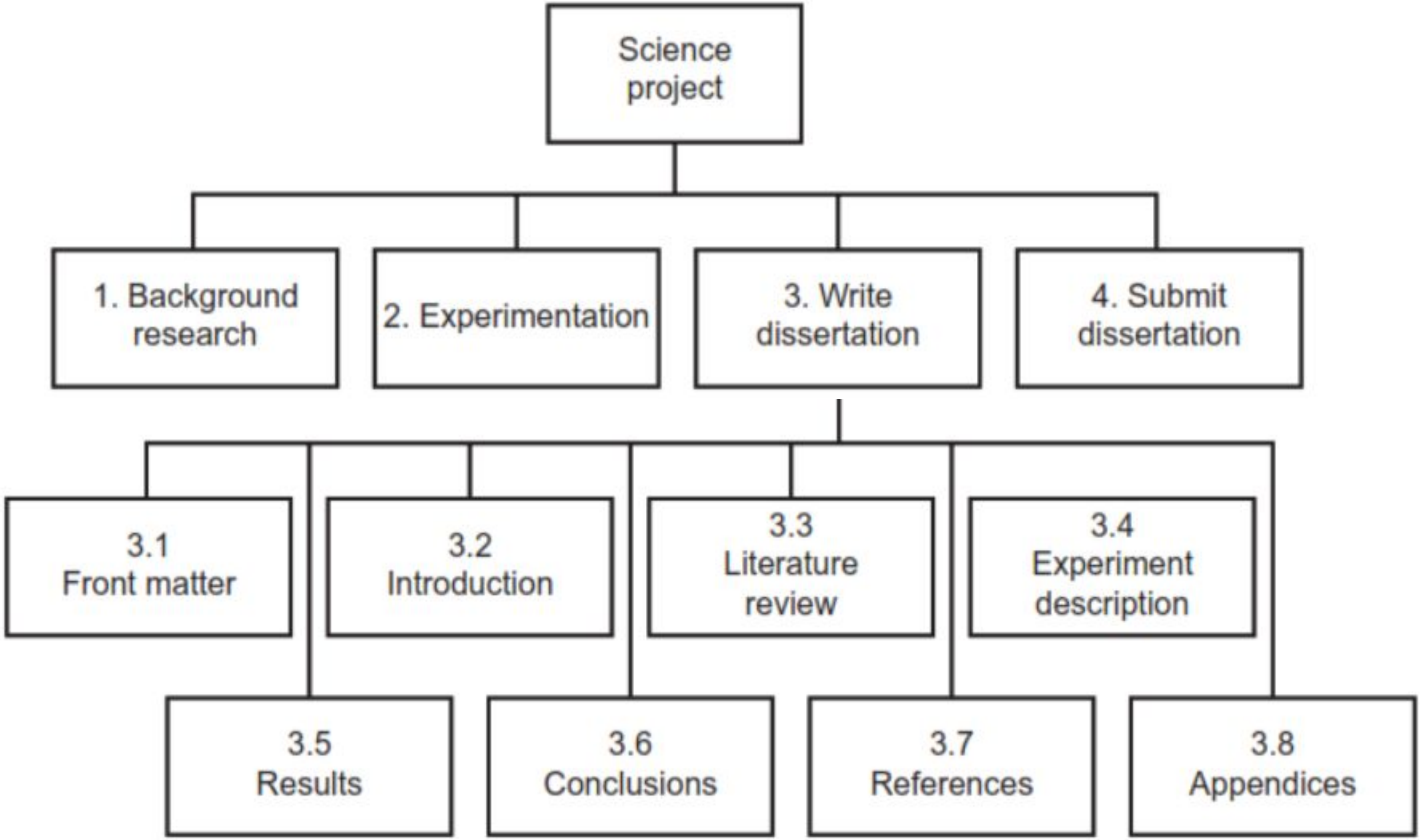
Karl Breecher, 2017

EXAMPLES

Tree Structure

Is it enough

?



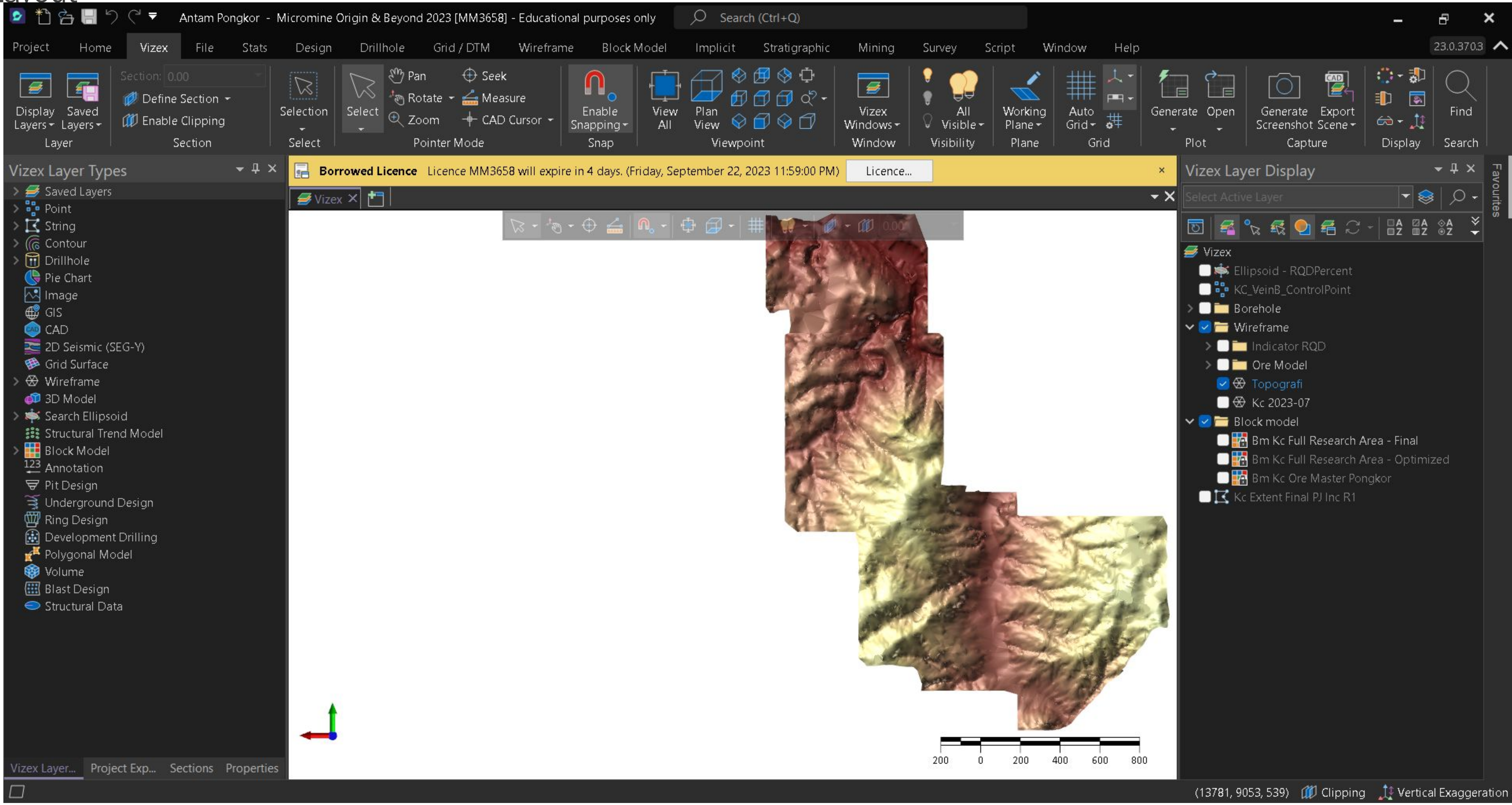
Karl Breecher, 2017

EXAMPLES - Mining

Software Layout

Is it enough

?



KEY INSIGHT

U need to know

Decomposition aids *collaboration*.

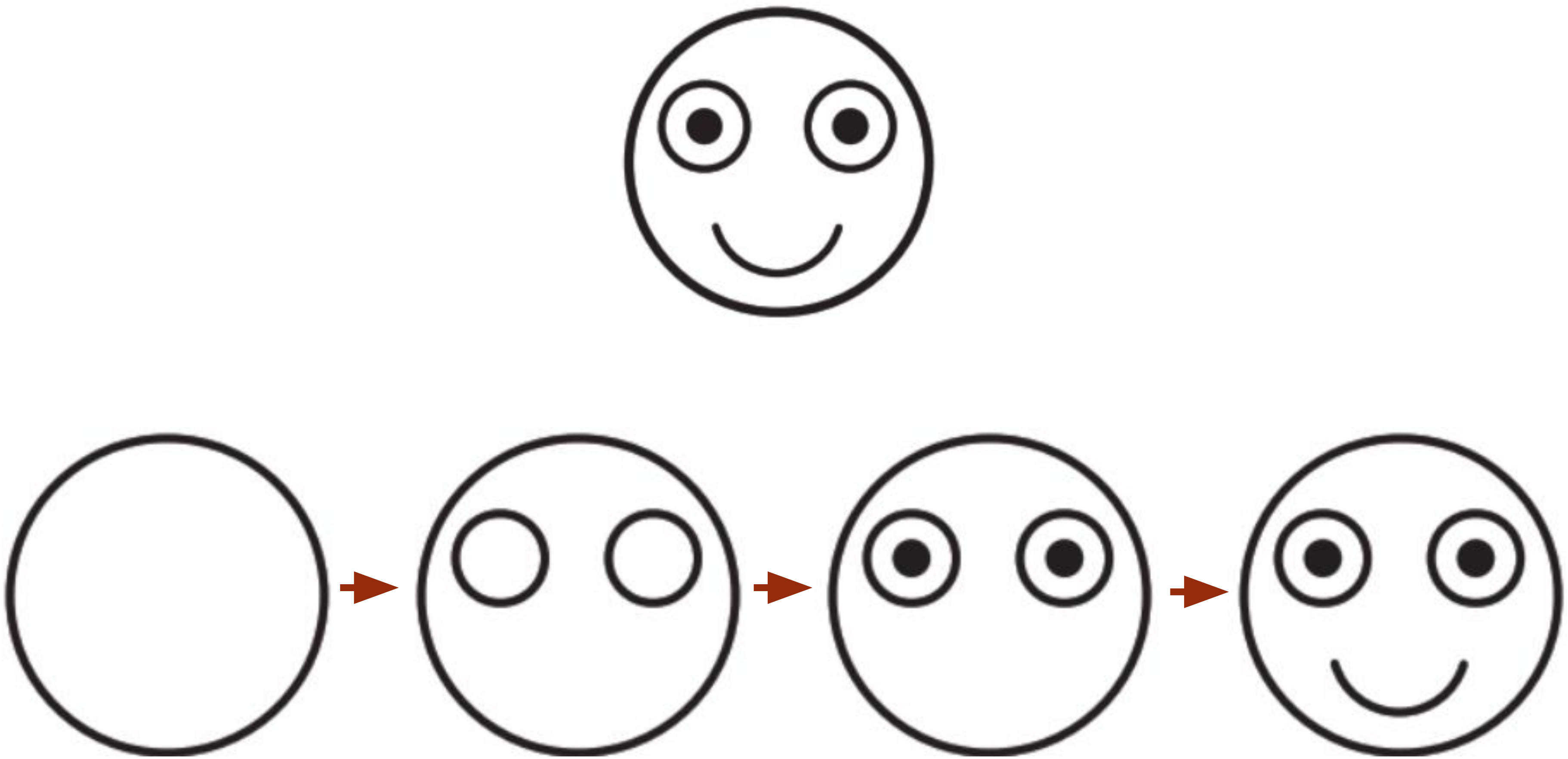
If you decompose a problem well (so that the subproblems can be solved independently),

then different people can work on different tasks, possibly in parallel.

Karl Breecher, 2017

KEY INSIGHT

Smiley Problem



Karl Breecher, 2017

KEY INSIGHT

Work Backwards

- 3.00 p.m. Arrive at Town Hall
- 2.55–3.00 p.m. Walk from Town Hall underground station to Town Hall
- 2.45–2.55 p.m. Take underground from City bus station to Town Hall underground station
- 2.30–2.45 p.m. Take bus from Main Street bus station to City bus station
- 2.25–2.30 p.m. Walk from house on Main Street to bus station
- 2.25 p.m. Departure time.

Karl Breecher, 2017

SEMESTER SYLLABUS

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term

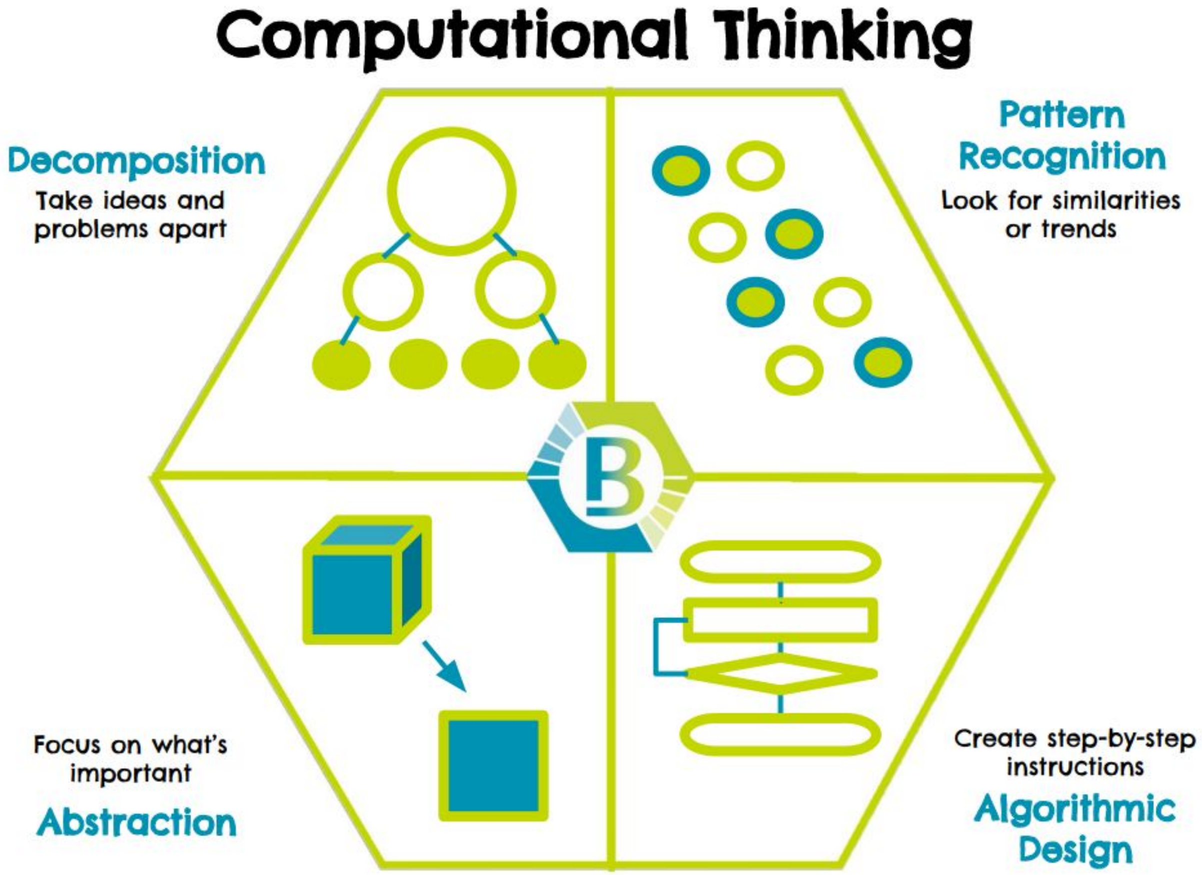
UNIVERSITAS TRISAKTI
PRODI TEKNIK PERTAMBANGAN

PATTERN & GENERALIZATION

MFU6207 - KOMPUTASI DASAR (*COMPUTATIONAL THINKING*)

7th session
1.5 hours

Lecturer :
Danu Putra, S.T., M.T.



www.wcpss.net

SEMESTER SYLLABUS

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

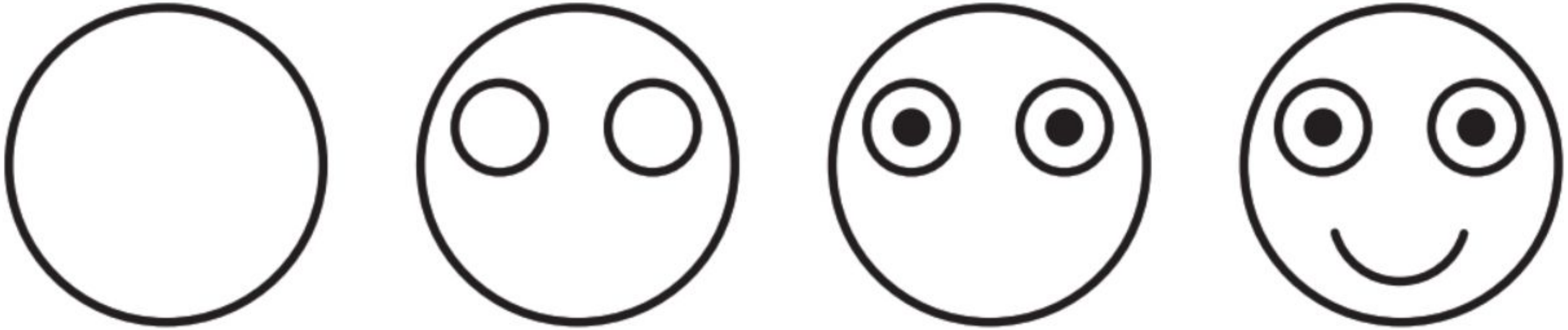
1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term

Why look for patterns?

If you don't automatically notice the patterns in a solution, you should make an effort to seek them out

Because they are the first step to making your solution more manageable and powerful.



1. Draw **circle** with radius 30 at position 50,50 with line thickness 2.
2. Draw **circle** with radius 6 at position 40,40 with line thickness 1.
3. Draw **circle** with radius 3 at position 40,40 filled black.
4. Draw **circle** with radius 6 at position 60,40 with line thickness 1.
5. Draw **circle** with radius 3 at position 60,40 filled black.
6. Draw red line from position 30,70 to position 70,70 with line thickness 1.

Karl Breecher, 2017

DEFINITION

Recognising simple patterns

Here’s one approach to spotting simple patterns:

Look for **nouns** that appear repeatedly. These could correspond to objects that your solution deals with.

Look for **verbs** that appear repeatedly. These could be operations that the solution carries out.

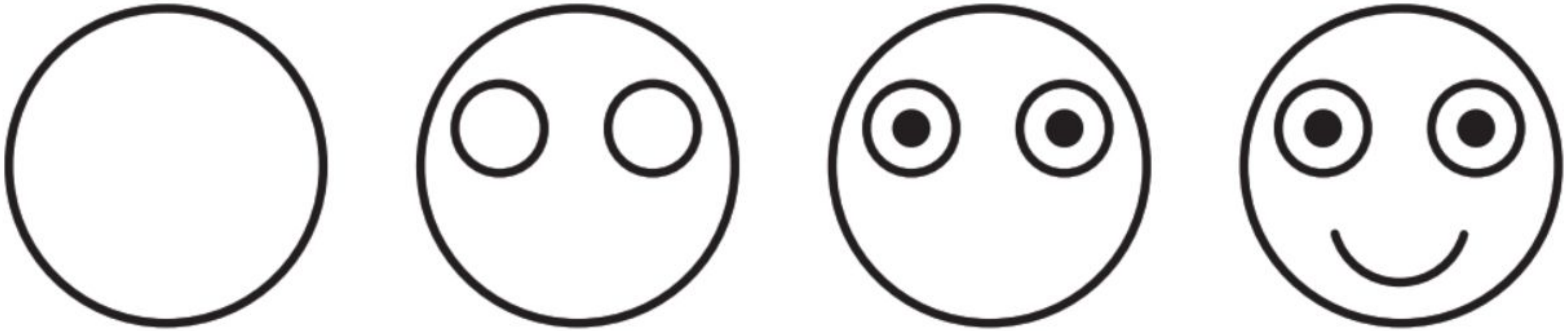
Look for **concrete descriptions**. These could probably be substituted by placeholders that vary in different situations. For example:

- **adjectives** (‘red’, ‘long’, ‘smooth’) which indicate properties of things and could be replaced by the property name (colour, size, texture);
- **actual numbers**, which could be replaced with variables.

Karl Breecher, 2017

DEFINITION

Recognising simple patterns



1. **Draw circle** with **radius 30** at **position 50,50** with **line thickness** 2.
2. **Draw circle** with **radius 6** at **position 40,40** with **line thickness** 1.
3. **Draw circle** with **radius 3** at **position 40,40 filled black**.
4. **Draw circle** with **radius 6** at **position 60,40** with **line thickness** 1.
5. **Draw circle** with **radius 3** at **position 60,40 filled black**.
6. **Draw red line** from **position 30,70** to **position 70,70** with **line** thickness 1.

Karl Breecher, 2017

DEFINITION

More complex patterns

Looking for larger and more complex patterns requires you to **expand your scope**, because you'll need to consider whole groups of instructions at once.

In brief, the guidelines for some of the more complex patterns are:

- Patterns among a **sequence** of instructions can be generalised into **loops**.
- Patterns among **separate groups** of instructions can be generalised into **subroutines**.
- Patterns among **conditionals or equations** can be generalised into **rules**.

Using loops

Generalization

Draw circle with radius 6 at position 40,40

Draw circle with radius 3 at position 40,40 filled black

Draw circle with radius 6 at position 60,40

Draw circle with radius 3 at position 60,40 filled black

Let coordinates = (40,40), (60,40)

For each coordinate x,y do the following:

  Draw circle with radius 6 at position x,y

  Draw circle with radius 3 at position x,y filled black

Karl Breecher, 2017

DEFINITION

More complex patterns

Looking for larger and more complex patterns requires you to **expand your scope**, because you'll need to consider whole groups of instructions at once.

In brief, the guidelines for some of the more complex patterns are:

- Patterns among a **sequence** of instructions can be generalised into **loops**.
- Patterns among **separate groups** of instructions can be generalised into **subroutines**.
- Patterns among **conditionals or equations** can be generalised into **rules**.

Using subroutines

Draw circle with radius r1 at position x,y

Draw circle with radius r2 at position x,y filled black

'Draw eye' is a subroutine (r1, r2, x,y):

Draw circle with radius r1 at position x,y

Draw circle with radius r2 at position x,y filled black

Generalization

Karl Breecher, 2017

More complex patterns

Looking for larger and more complex patterns requires you to **expand your scope**, because you'll need to consider whole groups of instructions at once.

In brief, the guidelines for some of the more complex patterns are:

- Patterns among a **sequence** of instructions can be generalised into **loops**.
- Patterns among **separate groups** of instructions can be generalised into **subroutines**.
- Patterns among **conditionals or equations** can be generalised into **rules**.

Using rules

'Draw eye' is a subroutine (r1, r2, x,y):

Draw circle with radius r1 at position x,y

Draw circle with radius r2 at position x,y filled black

Generalization

'Draw eye' is a subroutine (r, x, y):

Draw circle with radius r at position x,y

Draw circle with radius ½r at position x,y filled black

Karl Breecher, 2017

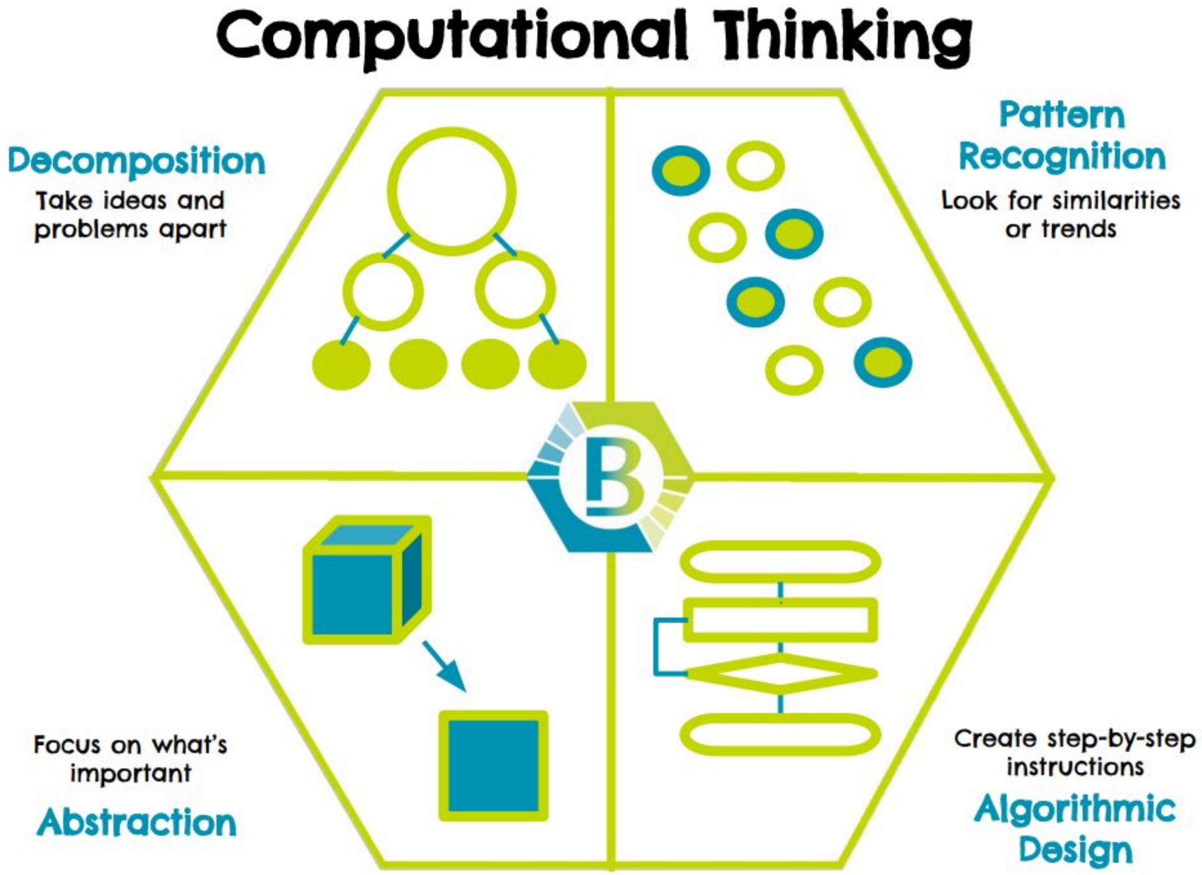
UNIVERSITAS TRISAKTI
PRODI TEKNIK PERTAMBANGAN

ABSTRACTION

MFU6207 - KOMPUTASI DASAR (*COMPUTATIONAL THINKING*)

8th session
1.5 hours

Lecturer :
Danu Putra, S.T., M.T.



www.wcpss.net

SEMESTER SYLLABUS

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term

The Importance

a way of ***expressing an idea in a specific context*** while at the same time ***suppressing details irrelevant*** in that context.

The essence of abstractions is ***preserving information that is relevant*** in a given context, and ***forgetting information that is irrelevant*** in that context. (Guttag, 2013)

Abstraction is a key feature of both computer science and computational thinking. Some have gone so far as to describe computer science as '***the automation of abstraction***. (Wing, 2014).

Karl Breecher, 2017

Why?

What we want : solve real-world problems using computers

The problem :

- the real world is **messy**, filled with lots of noise and endless details
- we're **not superbeings**; we're an intelligent species of ape with the ability to keep about seven or so pieces of information in working memory at any one time

How : Instead, we create **models** of the real world and then reason about the problem via these **models**

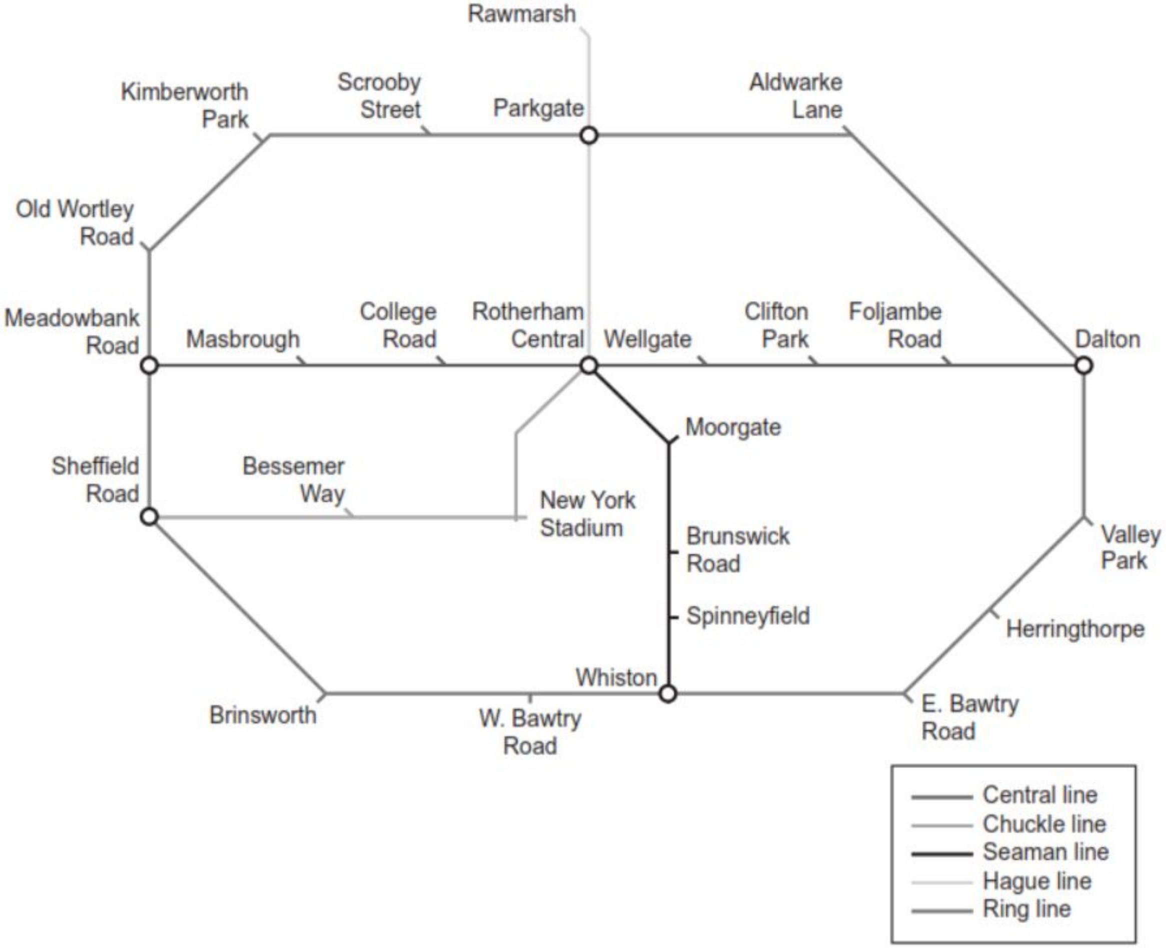
Karl Breecher, 2017

How to input this to computer ?

THE EXAMPLES

General

Real map



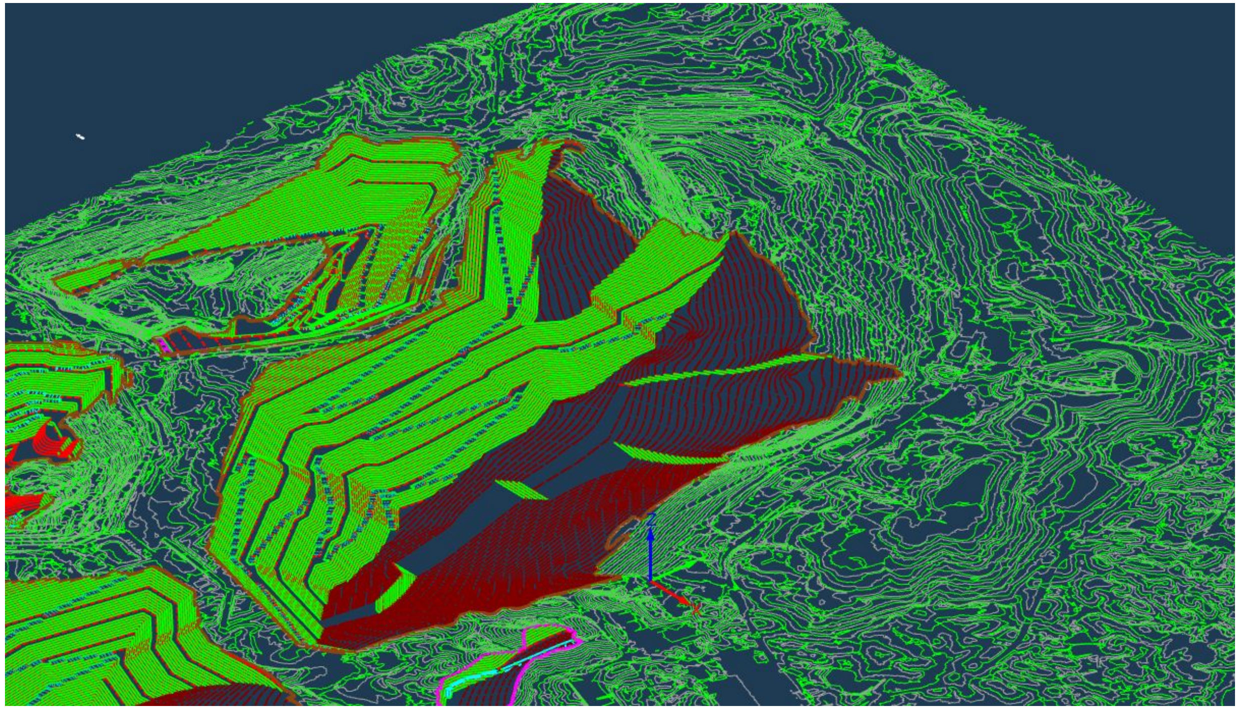
Simple version map

Karl Breecher, 2017

THE EXAMPLES

Mining

Photo



Design

Christina, 2023

THE EXAMPLES

Mining - Slopes

Karl Breecher, 2017

CAUTIONS

Treat Abstraction Carefully!

Abstractions may distract from reality

The reason you should be cautious is that abstractions are **a way of avoiding details**, either by suppressing or deferring them

Putting abstractions to use

It is **instantiated** and this requires attention to **details**

Leaking Details

No complex abstraction is perfect and all of them will likely leak at some point.

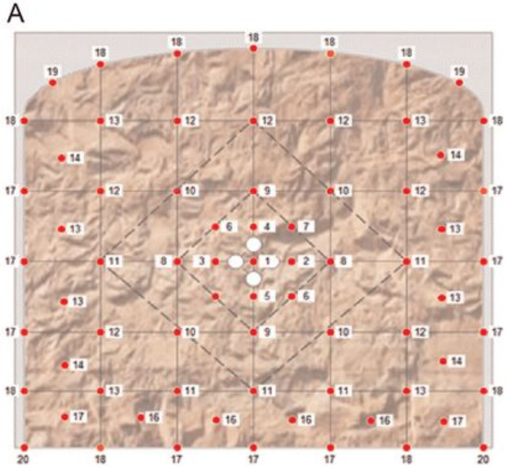
The same has been happening in science for centuries

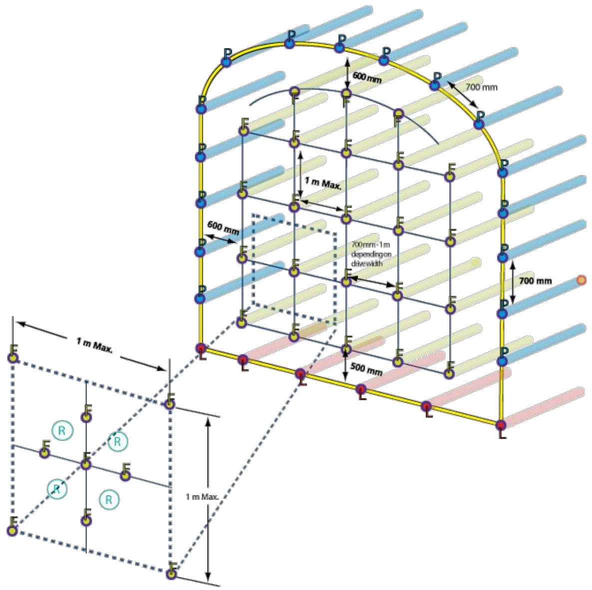
Details that turn out to be important need to be brought explicitly into the model.

Karl Breecher, 2017

CAUTIONS

Examples





Karl Breecher, 2017

SEMESTER SYLLABUS

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term

Assignment T4

Jelaskan yang anda ketahui mengenai modeling dan eror! (sampaikan sitasi)

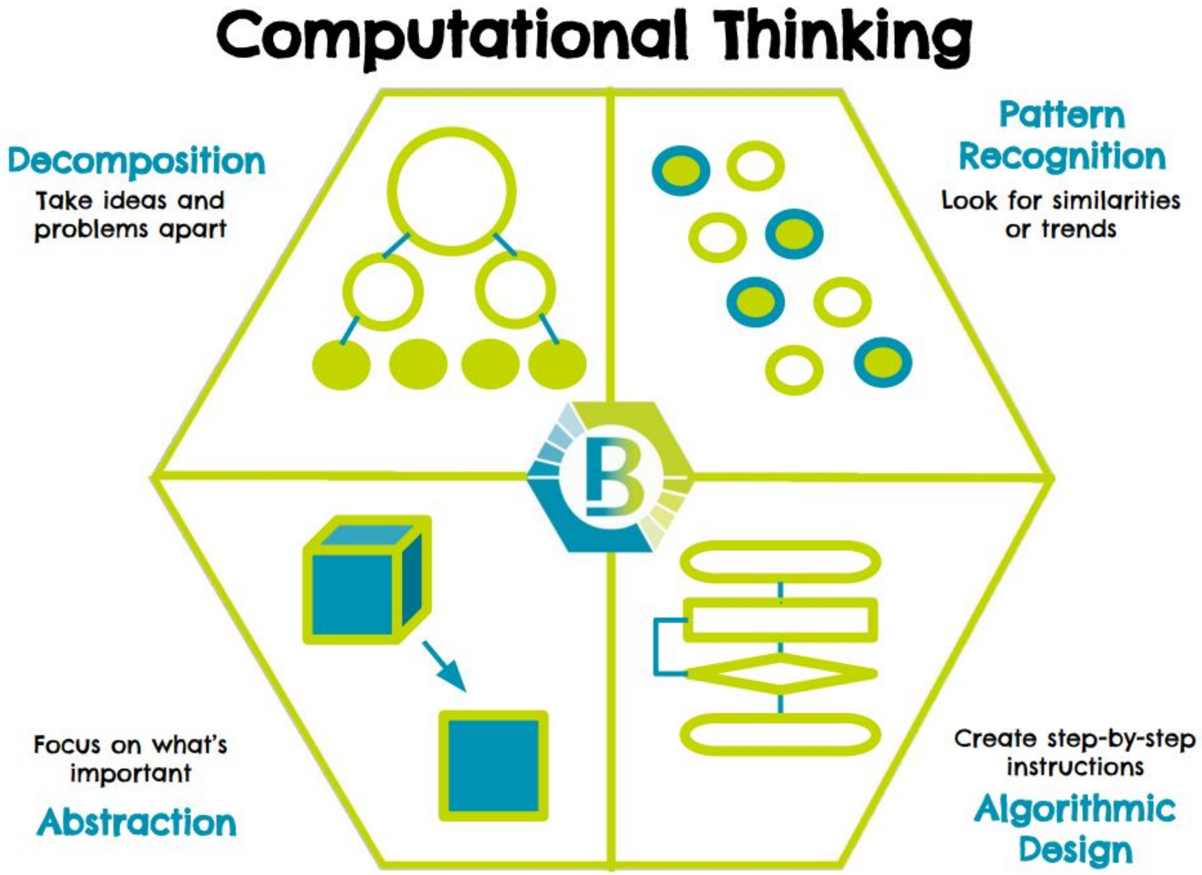
UNIVERSITAS TRISAKTI
PRODI TEKNIK PERTAMBANGAN

MODELING

MFU6207 - KOMPUTASI DASAR (*COMPUTATIONAL THINKING*)

6^{th} session
1.5 hours

Lecturer :
Danu Putra, S.T., M.T.



www.wcpss.net

SEMESTER SYLLABUS

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term

DEFINITION

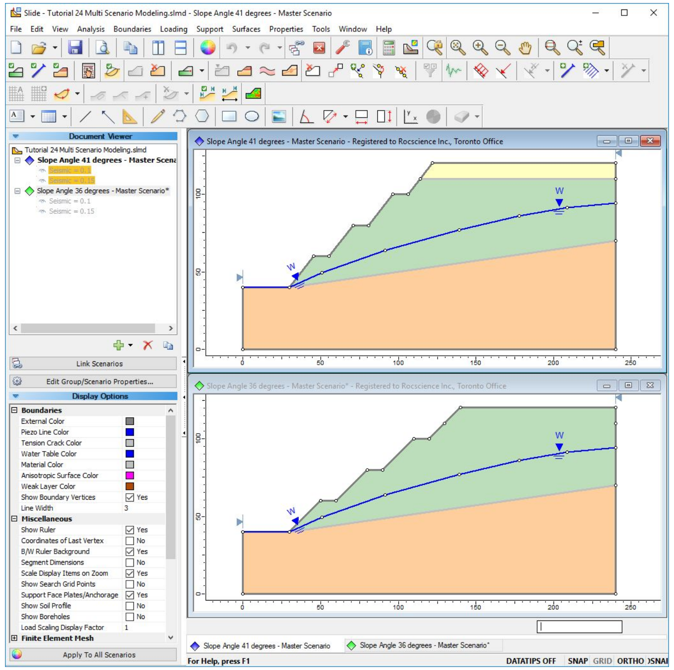
Definition – Motivation - Basic

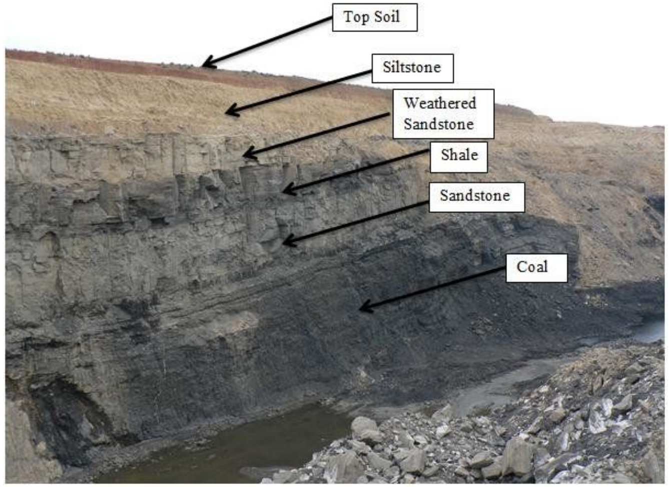
As **representations of real-world things that ignore certain details**, models are a type of abstraction and an important one in CS

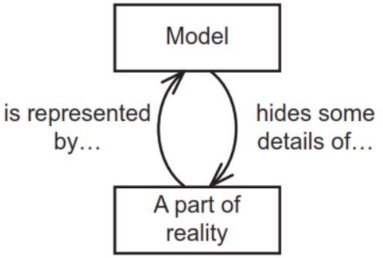
By showing a simplified view of things in your problem, models help you to **improve your understanding** because they allow you to **focus only on the relevant parts**

a model shows the entities in your solution and the relationships between them

- **entities**: the core concepts of the system
- **relationships**: a connection that relates entities in the system



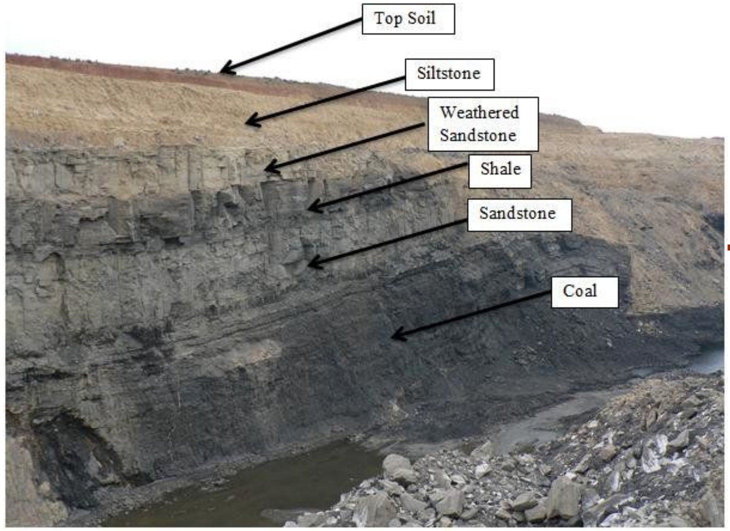




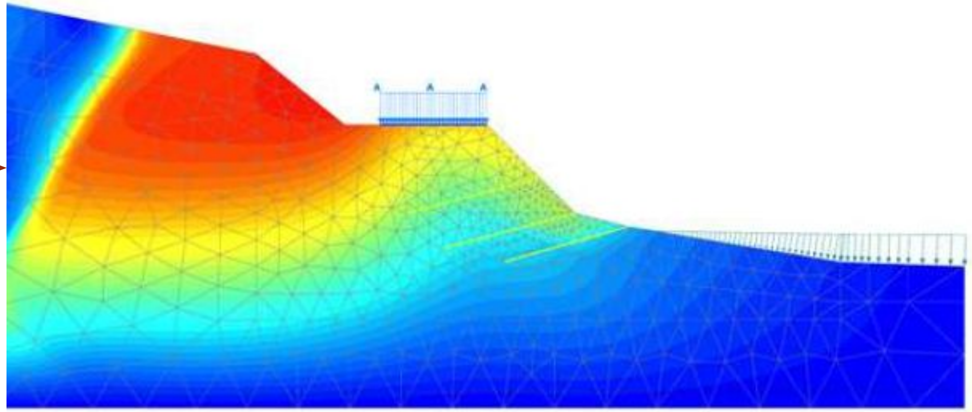
Karl Breecher, 2017

DEFINITION

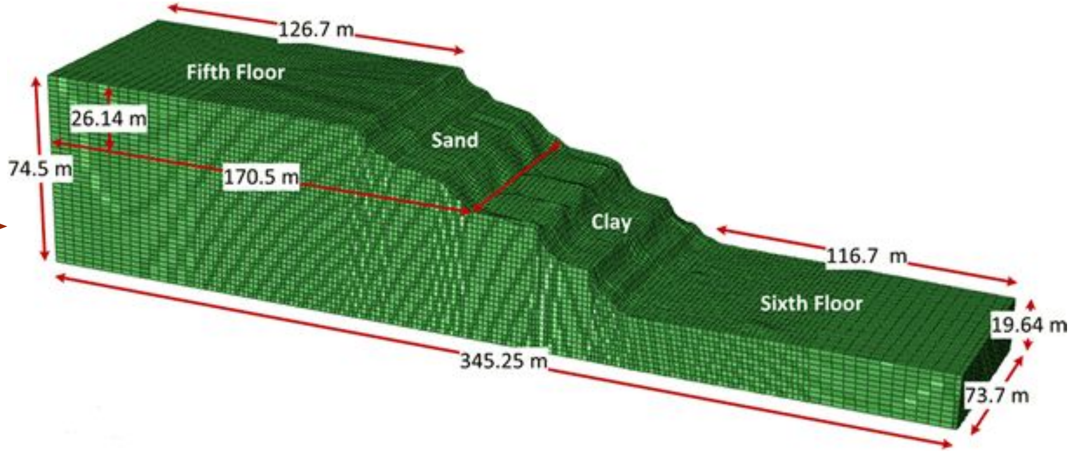
Definition – Motivation - Basic



2D



3D



Many types of models exist.

The same entities in your solution can be modelled over and over again in different ways by using different types of models

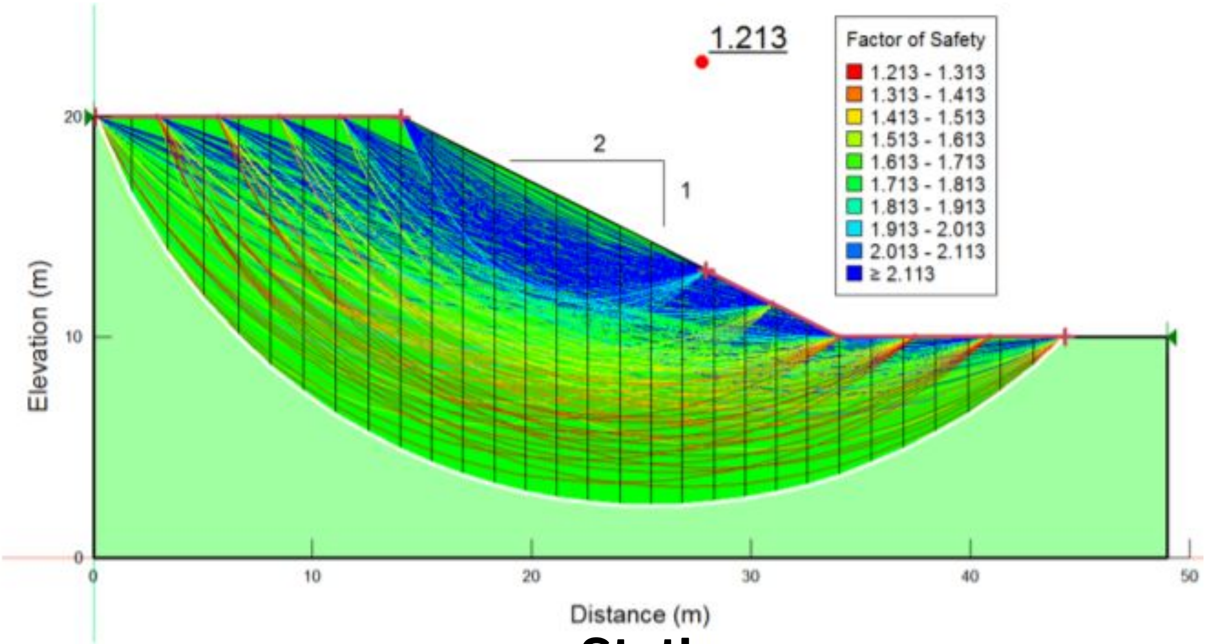
Karl Breecher, 2017

Definition – Motivation - Basic

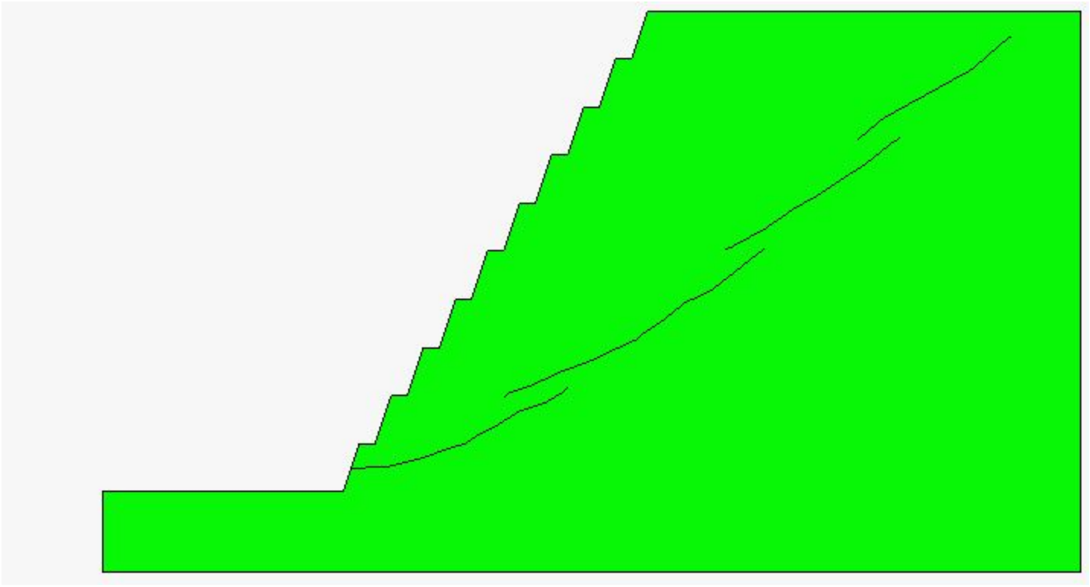
Static models give us a snapshot view of the system

depict entities and relationships **at a single point in time**

Dynamic model take into account that the world you're modelling changes over time



Static

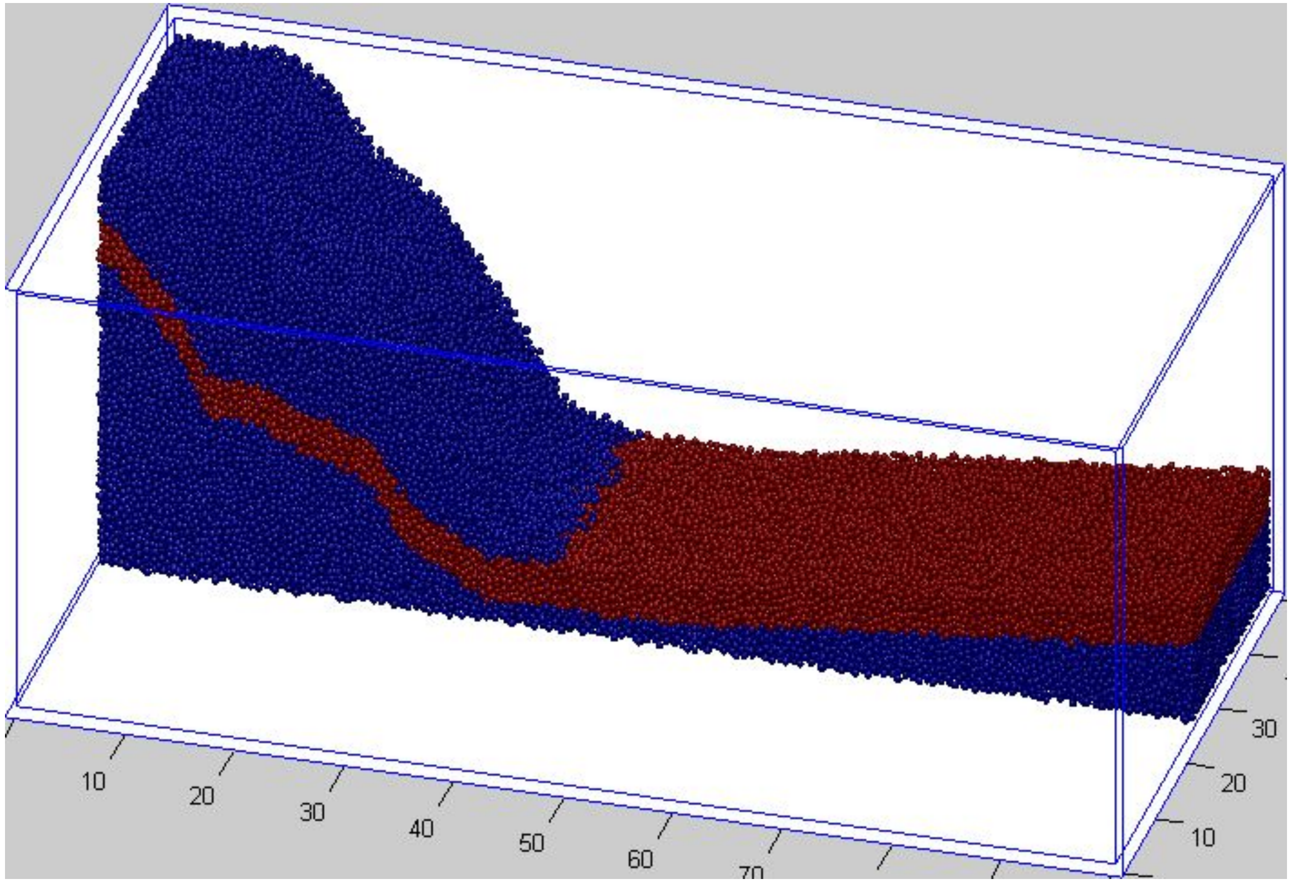


Dynamic

Karl Breecher, 2017; Plaxis; gifer.com

STATIC VS DYNAMIC MODEL

Definition – Motivation - Basic



Karl Breecher, 2017
Matdem.com

They always involve the following:

states: descriptions of the entities at specific points in time;

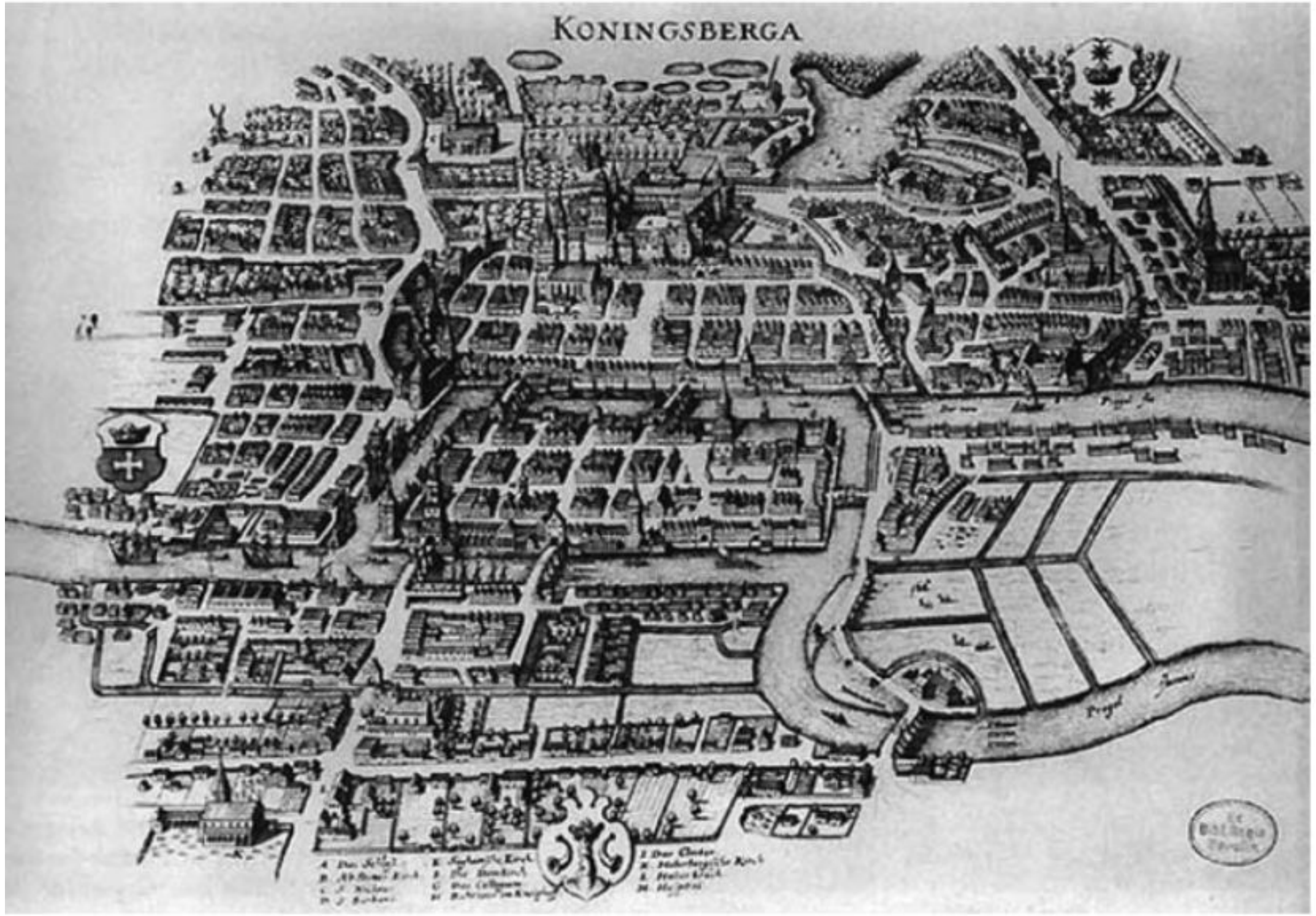
transitions: a change in state.

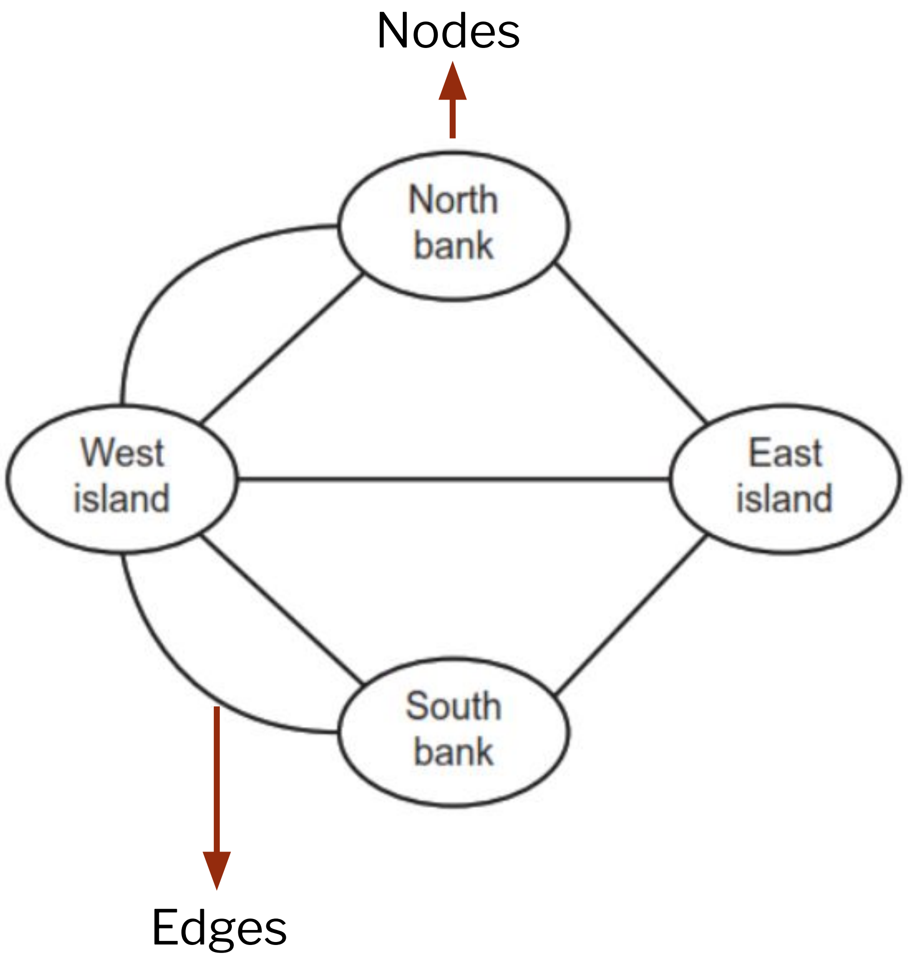
Although there are different varieties of dynamic models, they generally include some
or all of the following as well:

events: things that happen to trigger transitions;

actions: computations that are carried out as part of a transition.

Simplifiying and Explaining





Karl Breecher, 2017

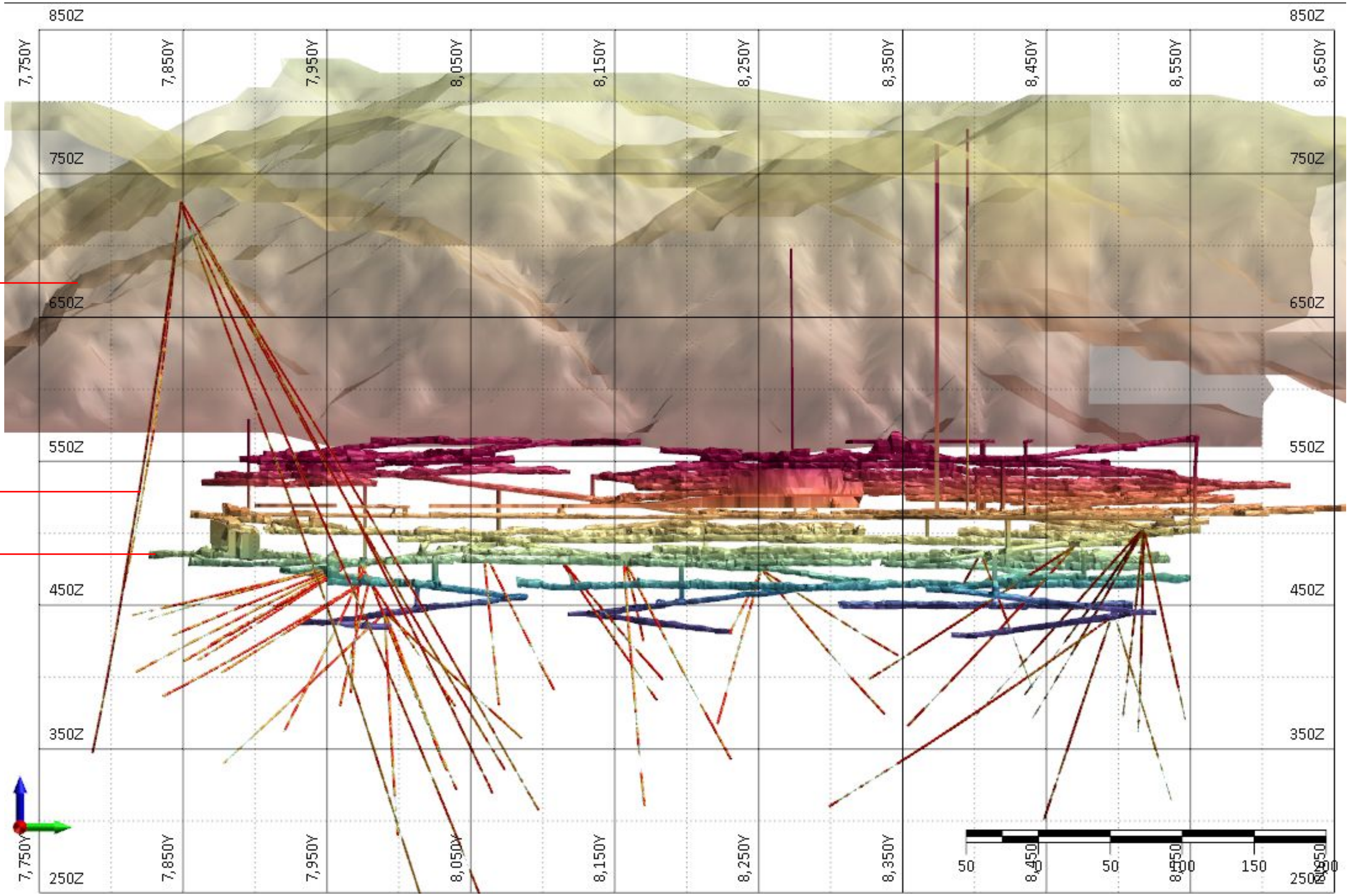
USES OF MODEL

Executing Representation

?

?

?



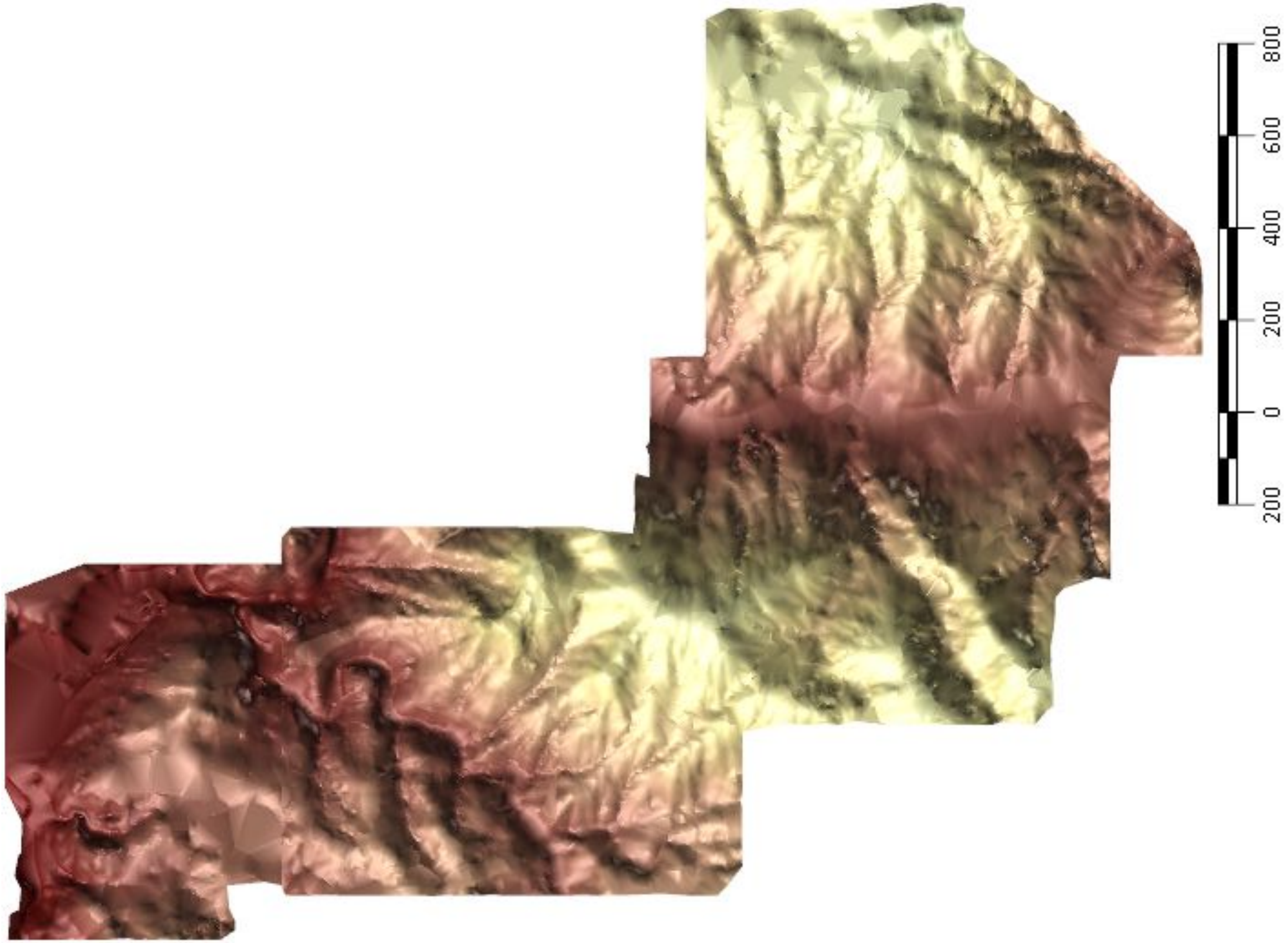
Danu, 2023

USES OF MODEL

Modeling Data

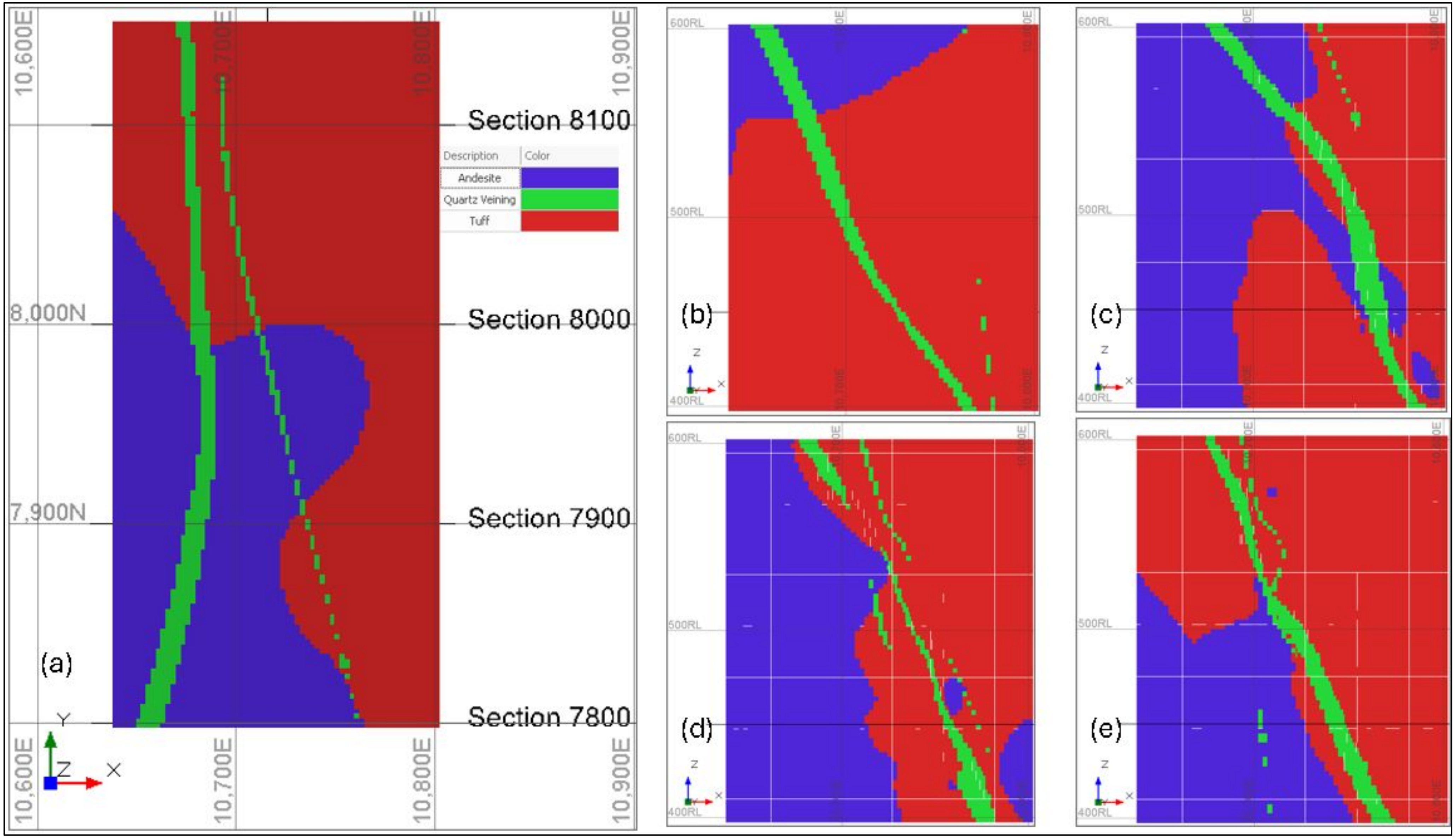
Node ID	X, Y, H	Feature
1	135, 483, 445	
2	250, 475, 457	
3	352, 430, 435	
4	153, 365, 465	
5	267, 330, 463	HOUSE
6	465, 335, 425	
7	050, 240, 452	
8	150, 050, 435	
9	240, 120, 432	
10	440, 240, 433	
11	430, 115, 447	
12	350, 050, 453	
A	352, 383, 419	STREAM
B	318, 297, 418	STREAM
C	250, 227, 421	STREAM
D	182, 225, 412	STREAM
E	095, 152, 409	STREAM
F	328, 187, 428	WELL

→



Danu, 2023

USES OF MODEL



Modeling method

Implisit modeling

Penyesuaian manual

Danu, 2023

THINGS TO REMEMBER

Purpose

- the model is useful
- **convey an idea**

Formality

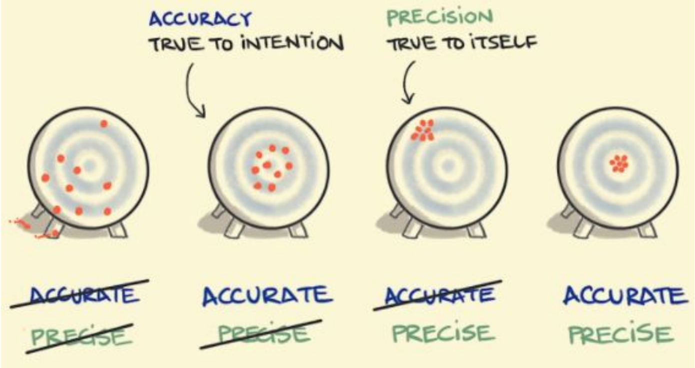
- some types of models have very specific rules about how to use them
- certain symbols might have specific meaning

Accuracy

- All models are wrong, but some are useful. (Box, 1987)
- what level of accuracy is acceptable?
- a model **may not be perfectly accurate**, but **it can be perfectly sufficient**

Precision

- values in the real world can be thought of as continuous
- no matter how precisely you measure a temperature (for example, 24.134 °C), you could always be more precise (24.1342 °C)
- when modelling continuous, real-world phenomena, we are usually constraining them.
- **it's up to you to judge how precise** your model needs to be. **Be mindful of the context and let it guide your judgement.**



Karl Breecher, 2017

NEXT SESSION

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term

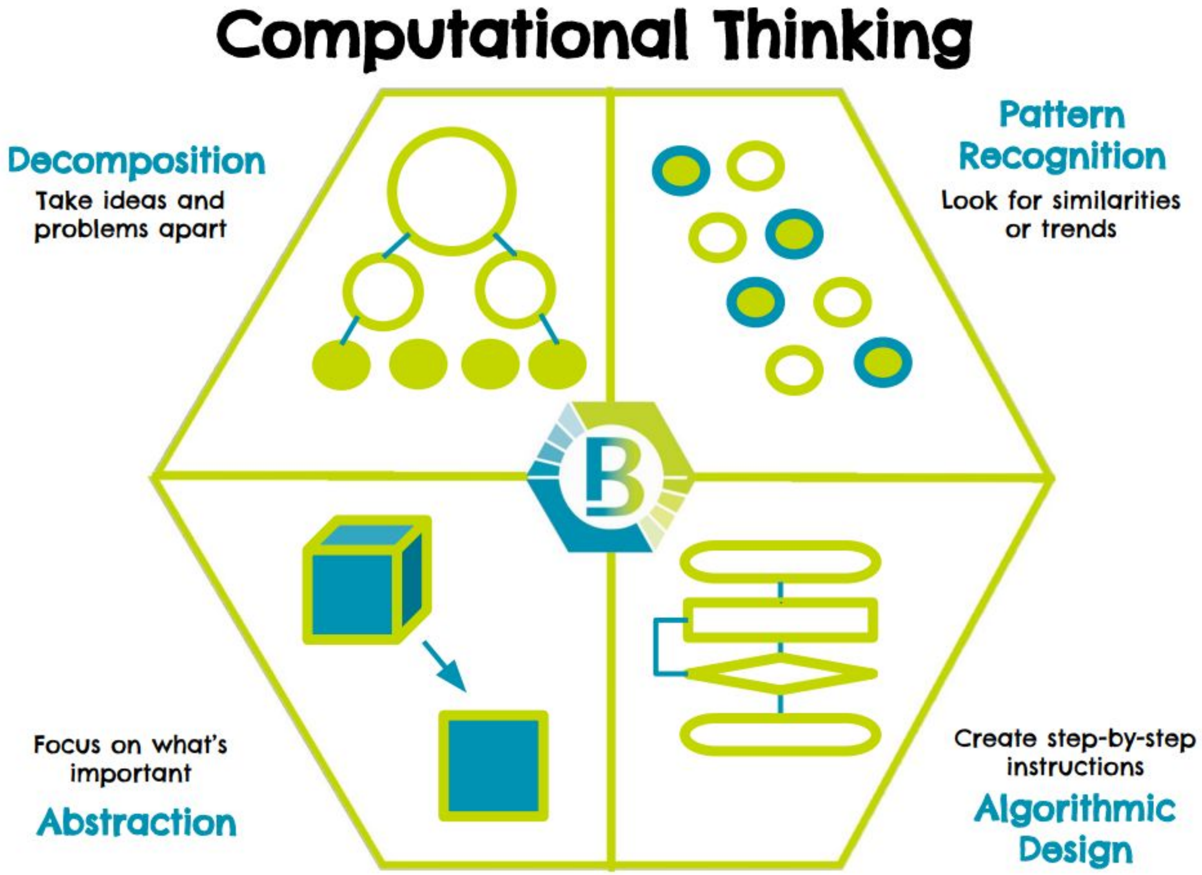
UNIVERSITAS TRISAKTI
PRODI TEKNIK PERTAMBANGAN

ERROR

MFU6207 - KOMPUTASI DASAR (*COMPUTATIONAL THINKING*)

6th session
1.5 hours

Lecturer :
Danu Putra, S.T., M.T.



www.wcpss.net

SEMESTER SYLLABUS

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term

KETIDAKPASTIAN (UNCERTAINTIES)

Definition – Motivation - Basic

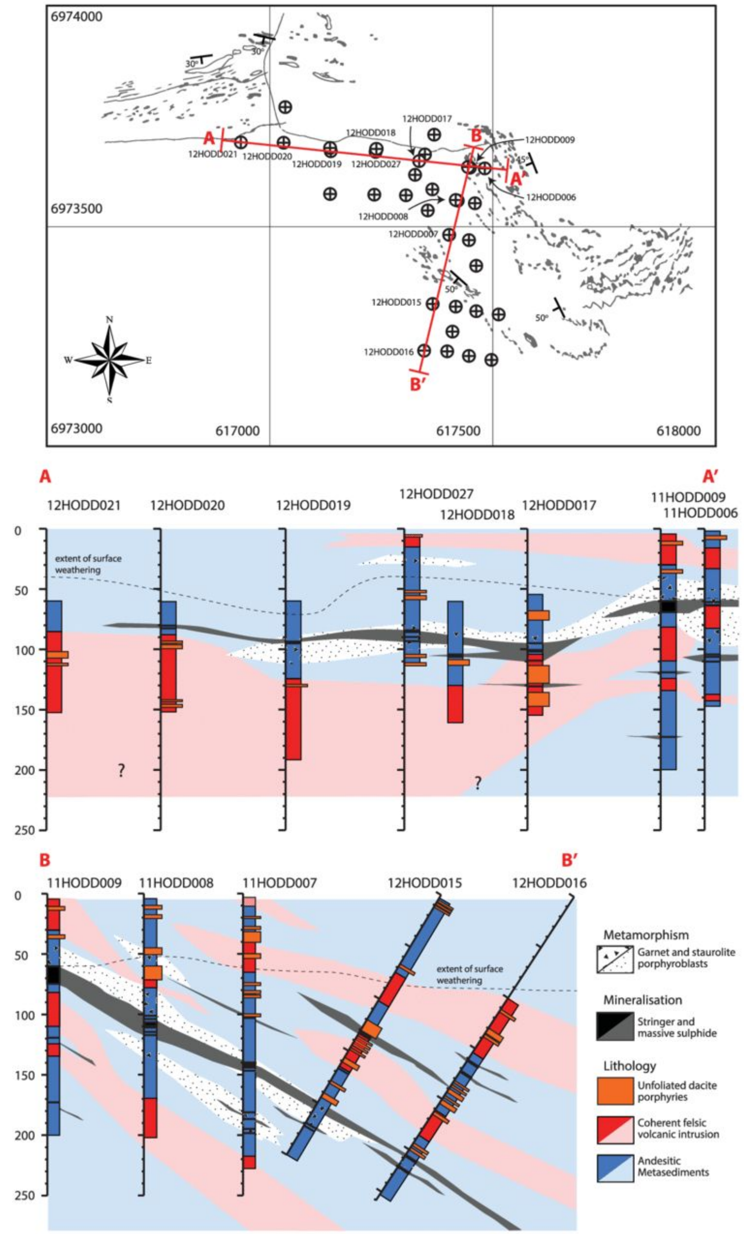
All geophysical, geological characterization and modeling of reservoir, involve **descriptions of reservoir properties** using **limited data** and thus **carry uncertainty in the predictions**.

Uncertainty analysis encompasses the quantification and reduction of uncertainty.

What stands in the way of predictability?

- **Logic**
- **Unknown physics**
- **Complicated physics**
- **Numerics**
- **Algorithms**

- **Statistics**
- **Experimental evidence**
- **Multi-scale Interactions**
- **Rocks Heterogeneity and limited knowledges**
- **Limited Data**
- **Methods application problems**
- **Modelling and simulations**
- **Interpretations**

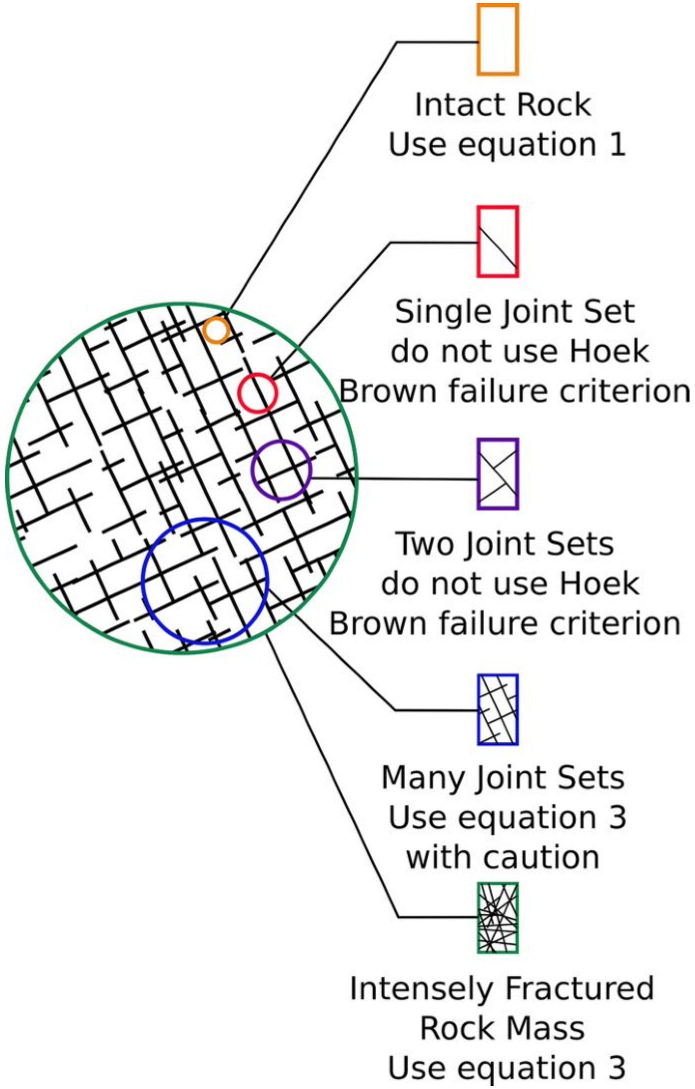


KETIDAKPASTIAN (UNCERTAINTIES)

Definition – Motivation - Basic

Geologic materials are **ubiquitously heterogeneous.**

Limited sampling results in uncertainty in spatial distribution of physical properties



ERRORS

Type Error

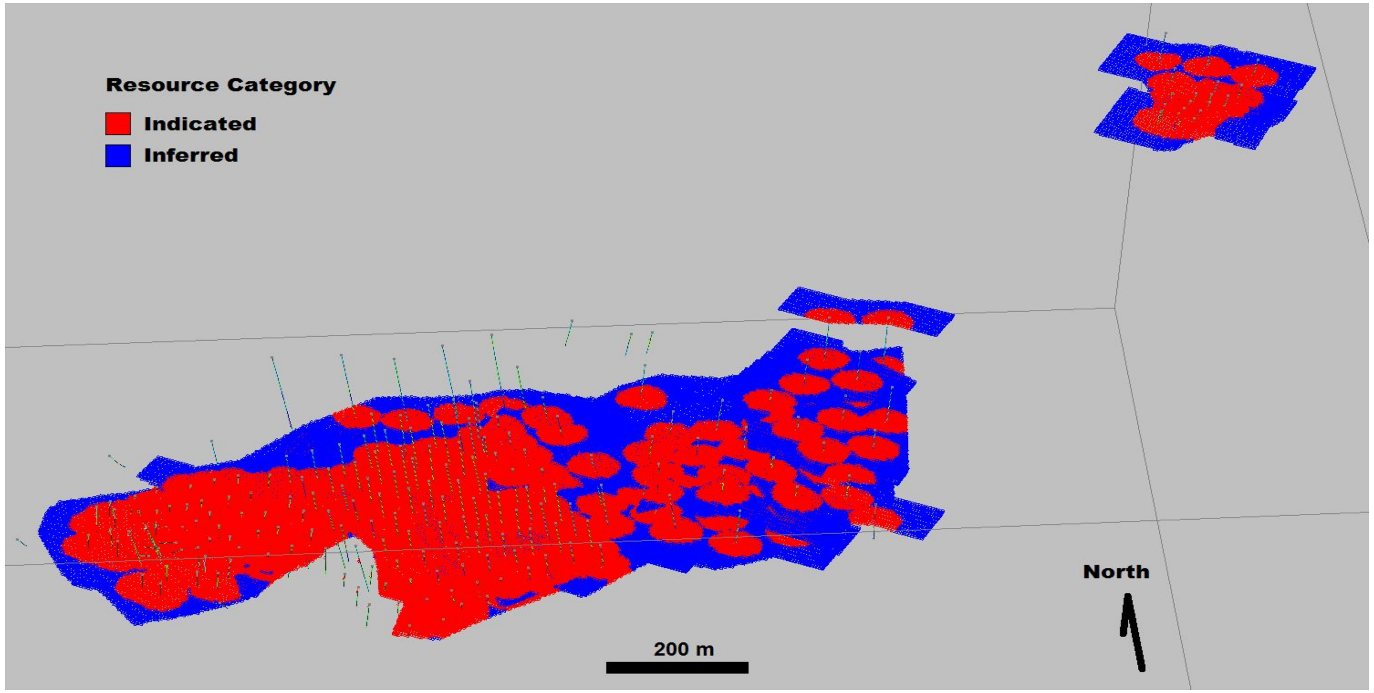
The difference between the measured and the true value.

- **Ilegitimate errors**
  - **Mistakes in technique** that result in blunders. You must use caution.
  - **Errors in computation** or calculation after the experiment.
- **Systematic errors**
  - An offset error (i.e. a measurement error) that persists over time.
    - Calibration can help to eliminate systematic mistakes.
    - Faulty equipment, such as a 3 percent high reading on an instrument
    - Observer bias—consistent or persistent human errors
  - Although this form of mistake **cannot be determined directly from the data**, **it can be estimated**.

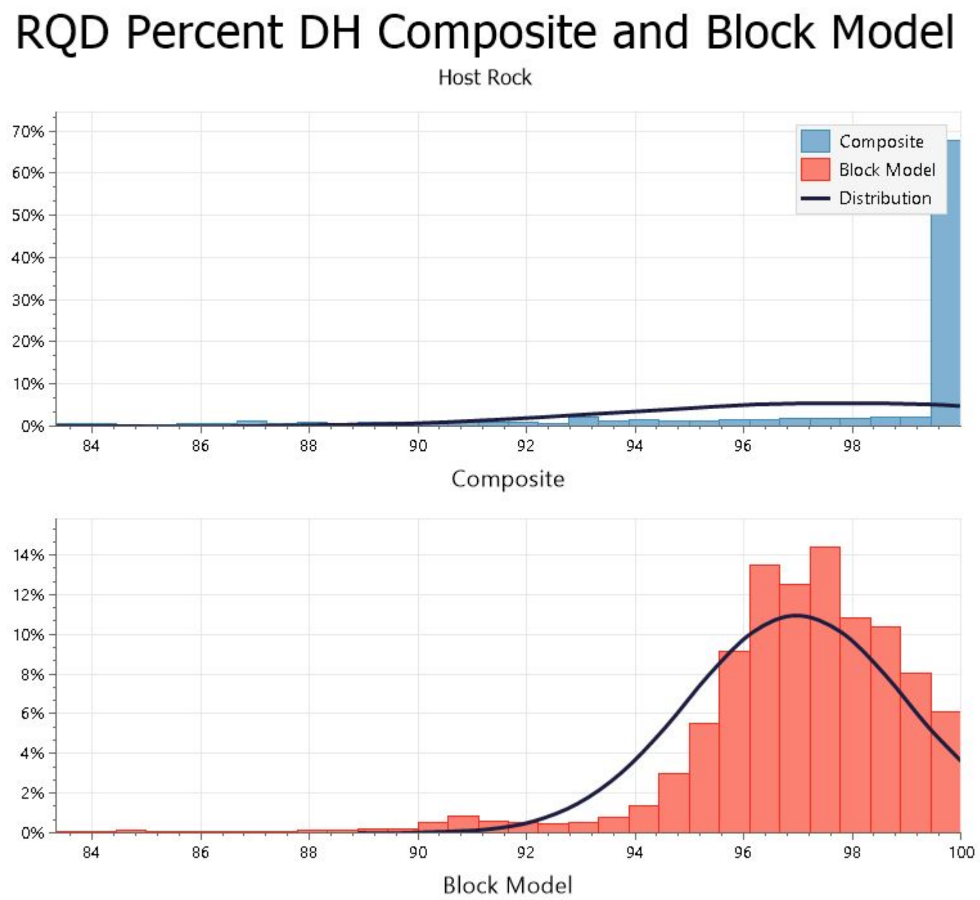
  Comparisons to theory or other experiments are used to arrive at a conclusion.

ERRORS

Type Error – Illegitimate error



Resource Classification



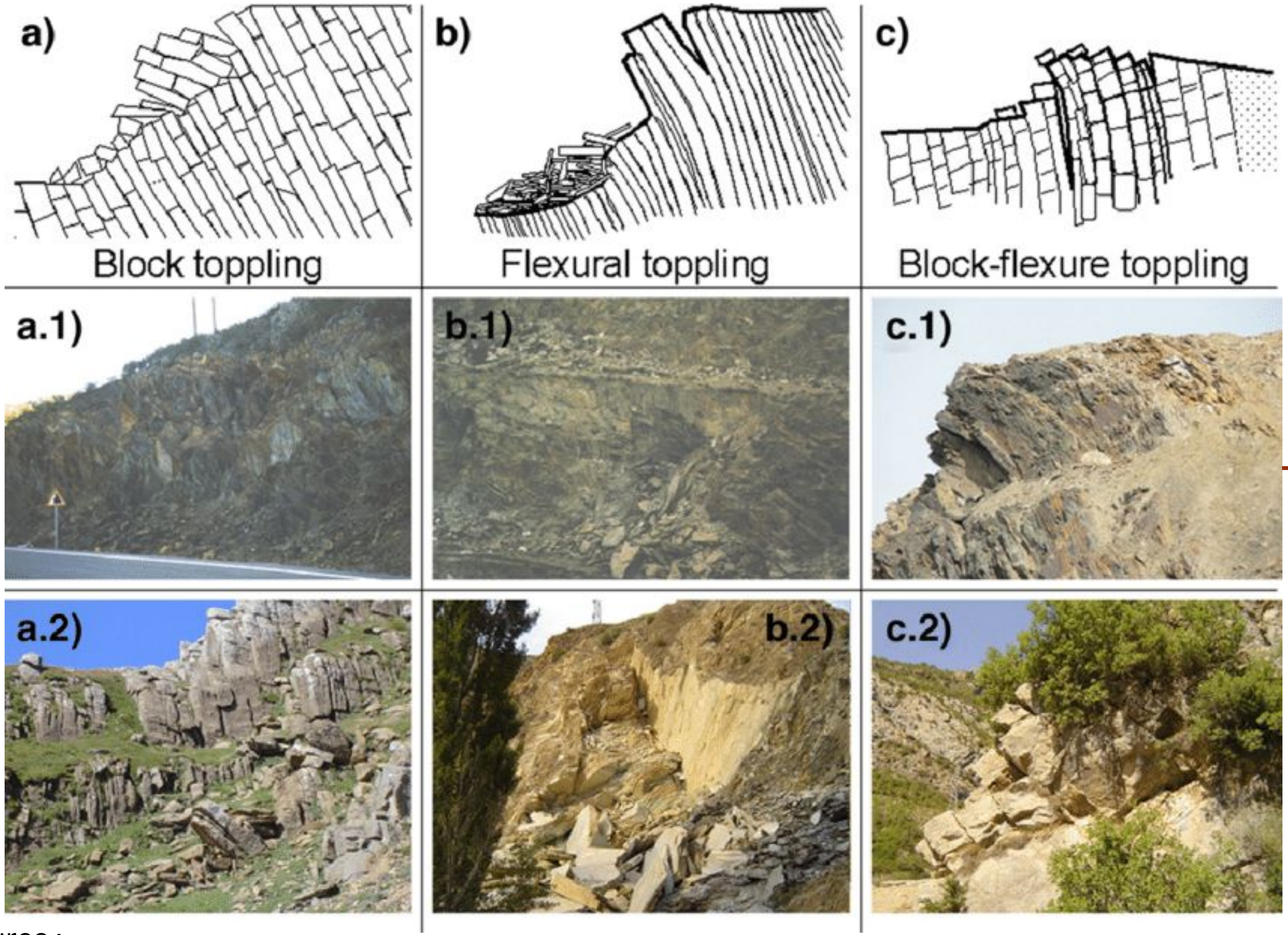
Estimate of Rock Parameter

Source : Putra, D. 2024

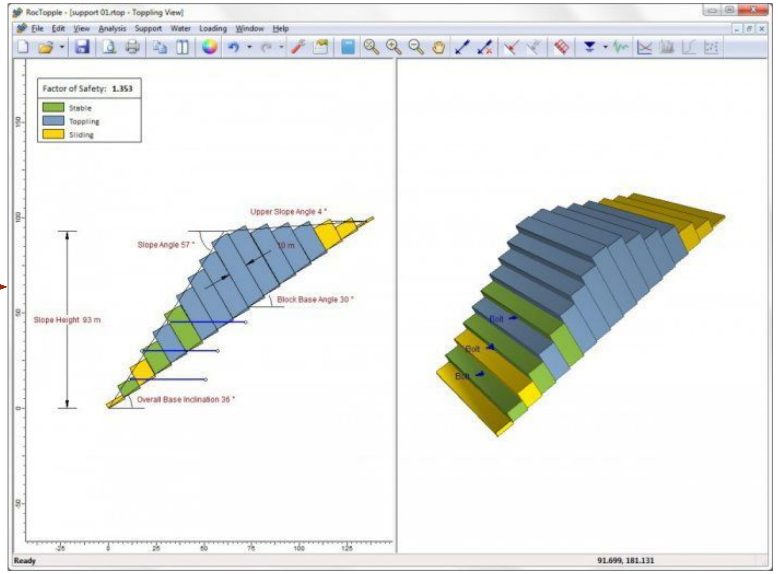
ERRORS

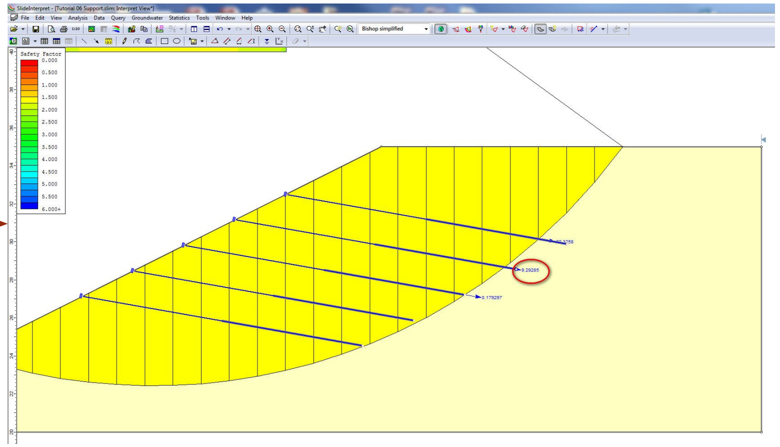
Type Error – Illegitimate error

Failure modeling



or



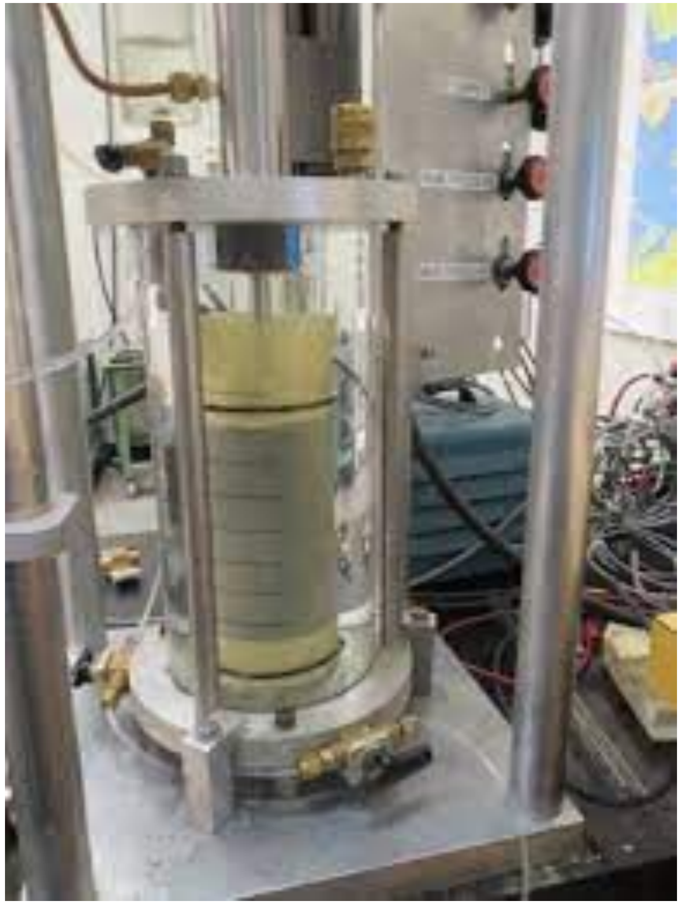


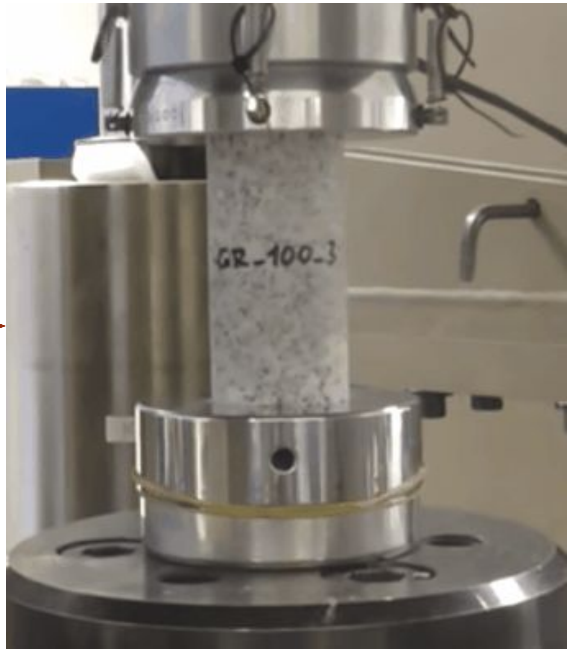
Source : rocscience

ERRORS

Type Error – Systematic error

Compressive Strength test



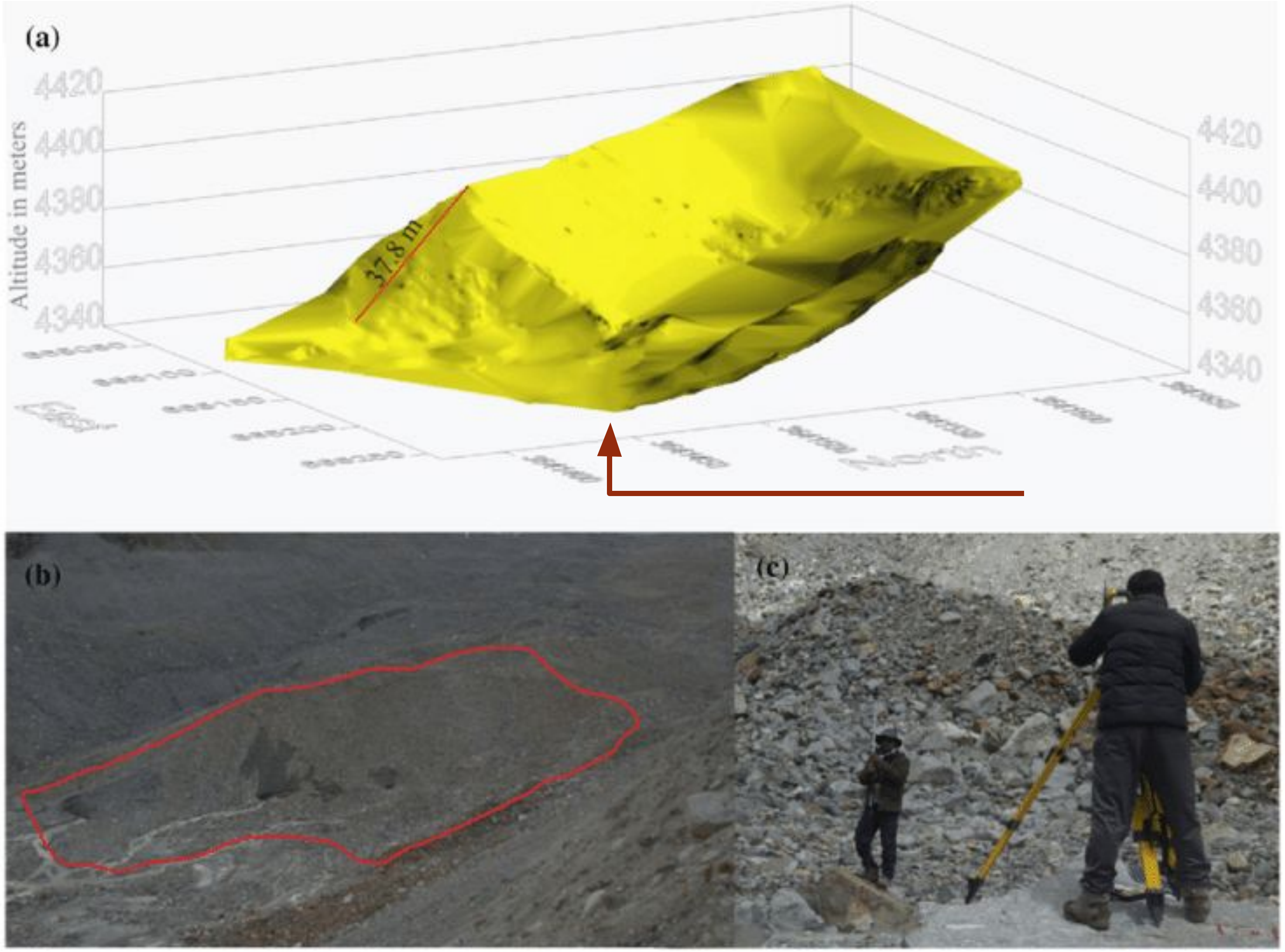


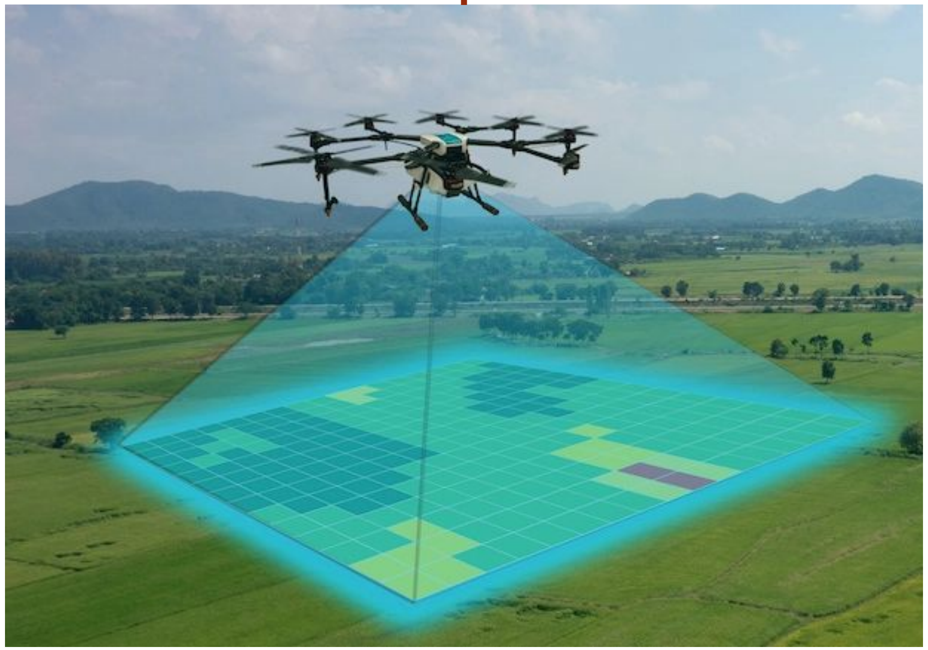
Source : civilblog.org; dreamcivil.com

ERRORS

Type Error – Systematic error

Mapping





Source : civilblog.org; dreamcivil.com

Type Error

- **Random, Stochastic or Precision errors:**
  - An error that causes readings to take random-like values about the mean value.
    - Effects of uncontrolled variables
    - Variations of procedure
  - The concepts of probability and statistics are used to study random errors. When we think of random errors we also think of repeatability or precision.

NEXT SESSION

1. Introduction
2. Apa itu computational thinking ?
3. Berpikir Logika (*Logical Thinking*) dan Algoritma (*Algorithmic Thinking*)
4. Berpikir Algoritma (*Algorithmic Thinking*) dan Penyelesaian Masalah *(Problem Solving)*
5. Dekomposisi (*Decomposition*) dan Abstraksi (*Abstraction*)
6. Model dan Error

Mid Term

1. Siklus Operasi Penambangan (Batubara)
2. Siklus Operasi Penambangan (Mineral)
3. Kebidangan dalam Pertambangan
4. Berpikir Komputasi dalam Software Penambangan
5. Case Study
6. Case Study
7. Case Study

Final Term