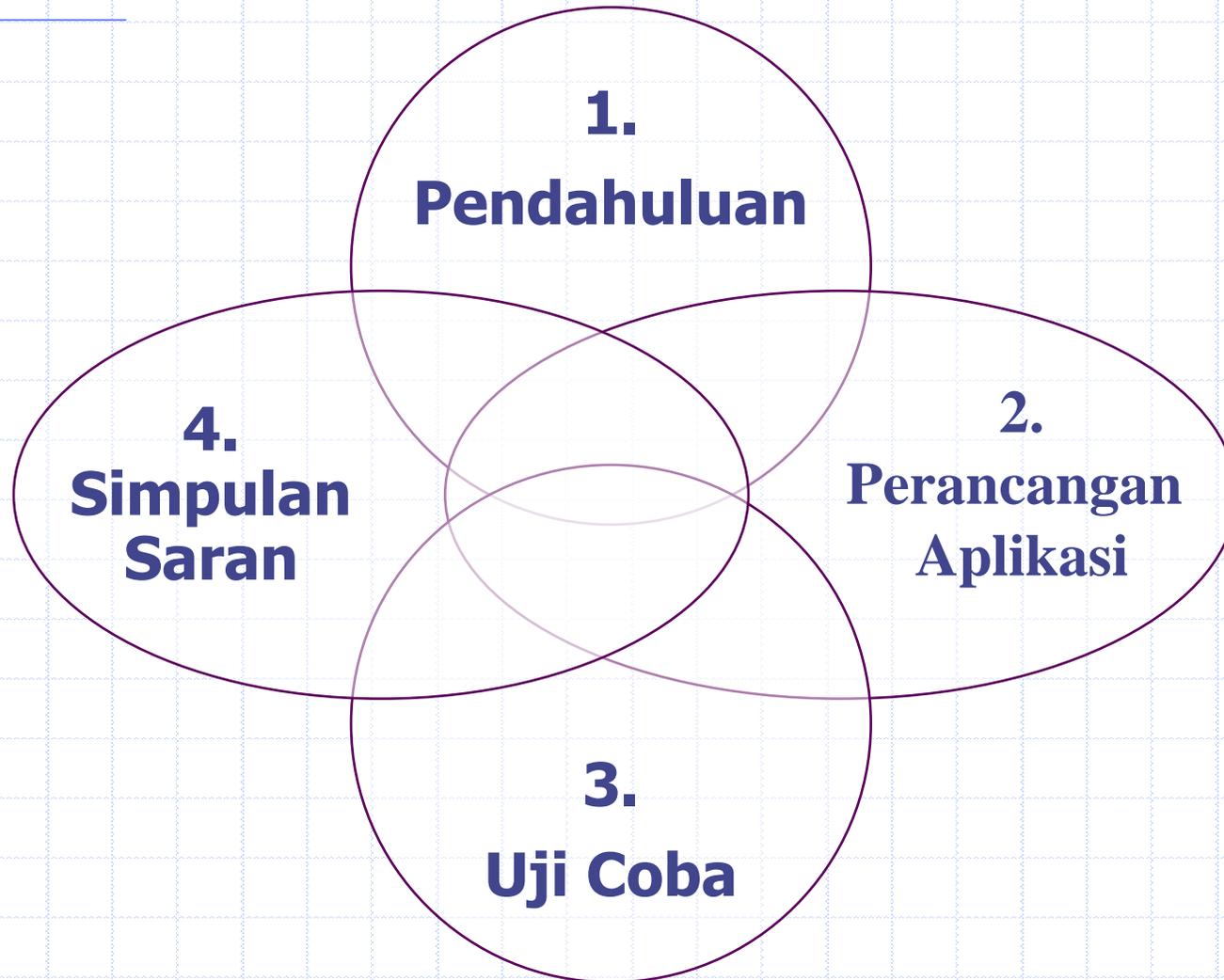

OPTIMASI *MIXED INTEGER LINEAR PROGRAMMING* PADA
PENYELESAIAN PERMASALAHAN PENGHINDARAN KONFLIK
MENGUNAKAN MODEL *VELOCITY AND ALTITUDE CHANGE*

Penyusun Tugas Akhir:

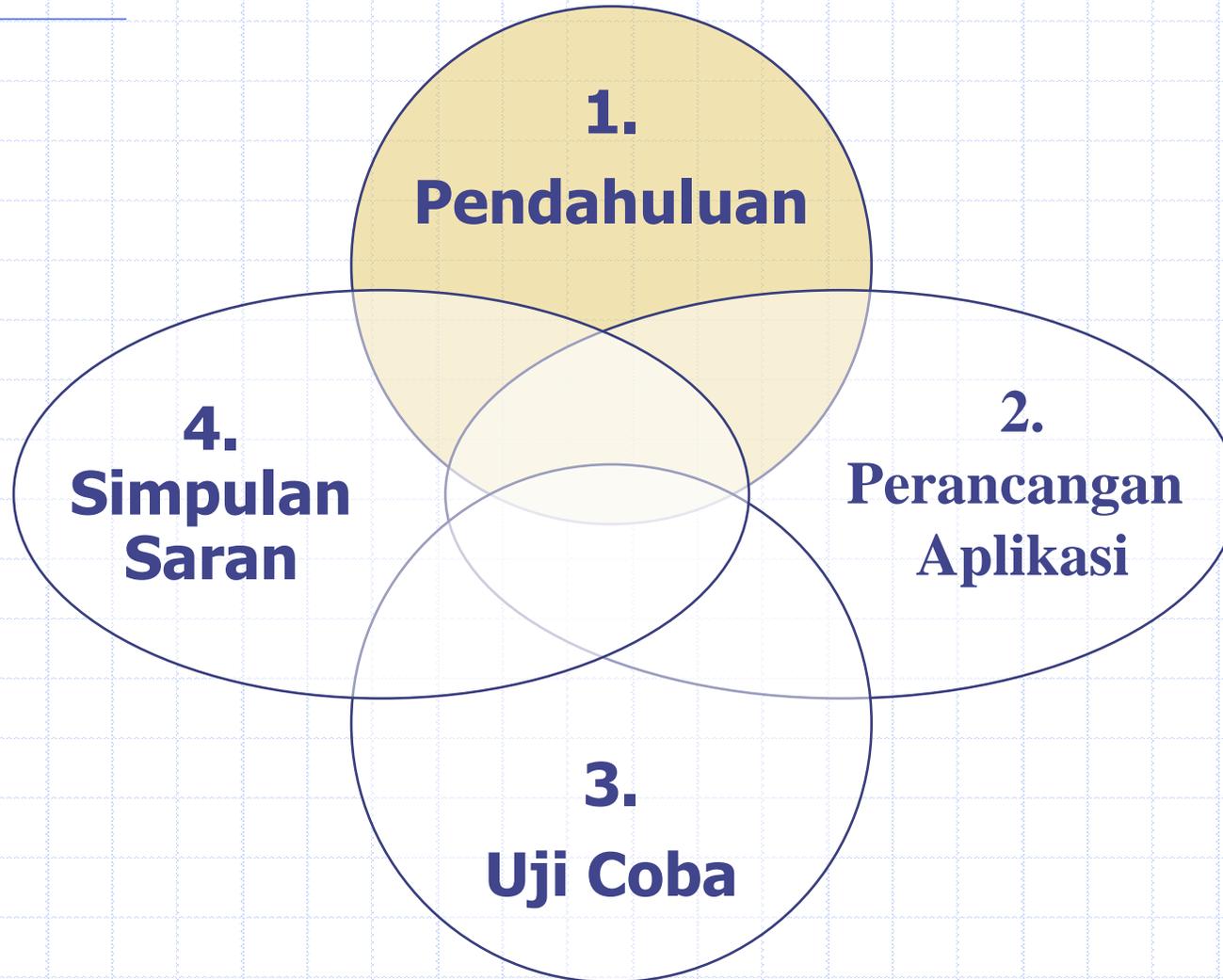
Izdihar Farahdina
(5112 100 191)

Dosen Pembimbing:

Victor Hariadi, S.Si., M.Kom.
Rully Soelaiman, S.Kom., M.Kom.



PENDAHULUAN



.:LATAR BELAKANG:.

- ◆ Ramainya pertumbuhan *traffic* di wilayah udara me-nambah kebutuhan yang harus diterapkan pada sistem *Air Traffic Control* (ATC)
- ◆ Sebuah sistem yang dapat menghindari pesawat dari konflik dan total perubahan yang tidak mengganggu *time flight*
- ◆ Kriteria konflik adalah 5 nm jarak horizontal minimum antara pesawat atau 1.000 ft sebagai jarak vertical minimum
- ◆ Permasalahan yang diangkat dalam tugas akhir ini adalah *Conflict Avoidance (CA)* dengan menggunakan model VAC yang diusulkan oleh Alonso et al[1][5]

.:TUJUAN DAN MANFAAT:.

- ◆ Tujuan dari tugas akhir ini adalah :
 - Melakukan desain serta analisis penyelesaian permasalahan CA dengan model VAC.
 - Melakukan analisis terhadap kinerja model VAC untuk penyelesaian permasalahan CA.

- ◆ Manfaat dari tugas akhir ini adalah :
 - membantu meminimalkan efek dari penghindaran konflik yaitu pesawat datang tidak sesuai jadwal.

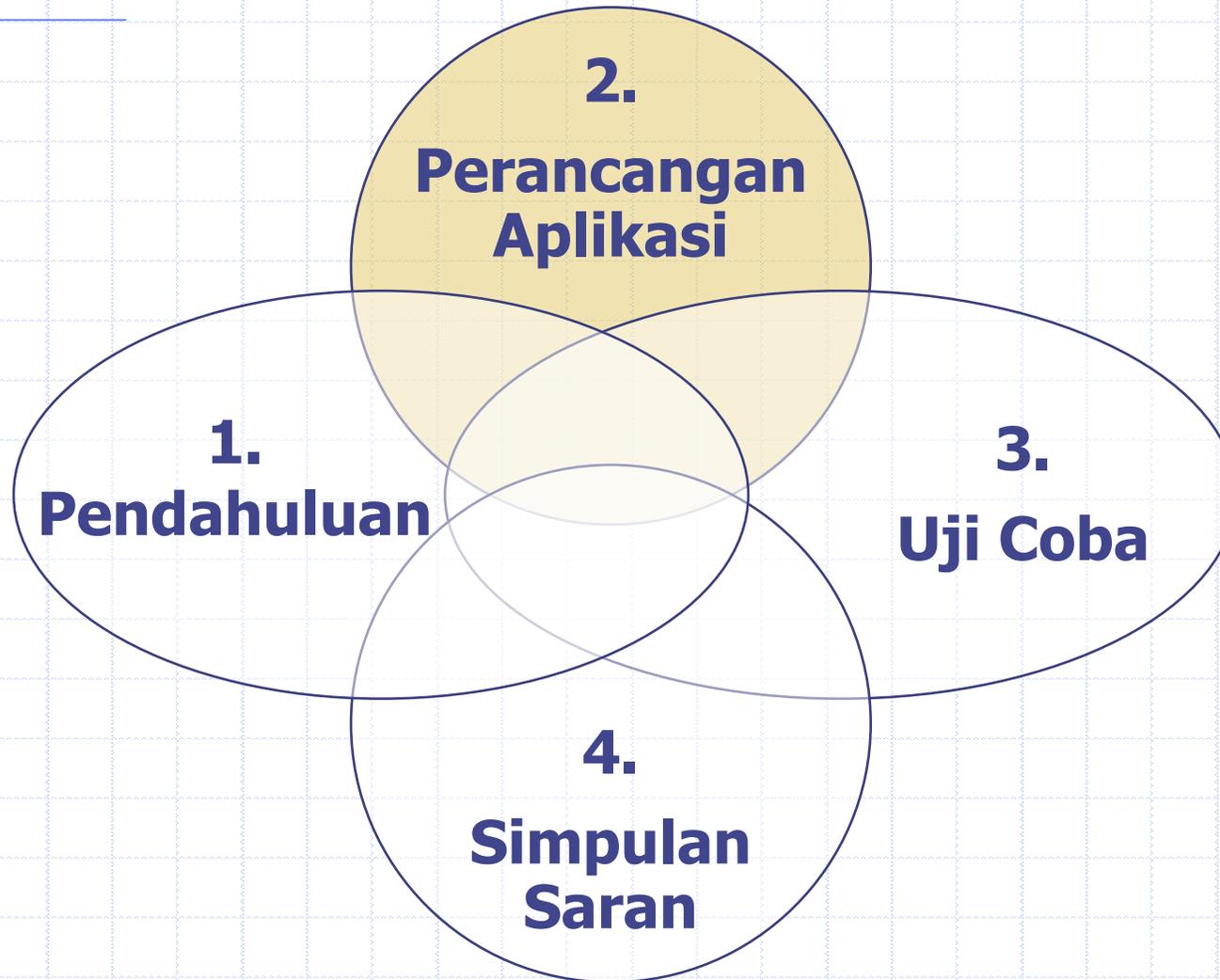
.:PERMASALAHAN:.

- ◆ Bagaimana menerapkan model VAC untuk mendapatkan kondisi pesawat yang tidak termasuk dalam kriteria konflik?
- ◆ Bagaimana mengimplementasi model VAC untuk menyelesaikan permasalahan penghindaran konflik?
- ◆ Bagaimana hasil uji coba model VAC untuk menyelesaikan permasalahan penghindaran konflik?

.:BATASAN PERMASALAHAN:.

- ◆ Data yang digunakan adalah penomoran pesawat(f), koordinat(x_f, y_f), sudut arah terbang(m_f^*), kecepatan (v_f), kecepatan minimal (\underline{v}_f), kecepatan maksimal(\bar{v}_f), *safety radius*(r_f), dan level ketinggian (z_f)
- ◆ *Safety radius* untuk setiap pesawat adalah sama, yaitu 2.5 nm.
- ◆ Implementasi dilakukan dengan bahasa pemrograman SAS.
- ◆ Penyelesaian masalah hanya memakai dua manuver yaitu kecepatan dan ketinggian. Hasil tidak mempertimbangkan faktor luar.
- ◆ Arah terbang dan kecepatan pesawat pada *time frame* t tetap untuk *time frame* selanjutnya, $t+1, t+2, t+3....t+n$.

PERANCANGAN APLIKASI



- Fungsi Obyektif : meminimalkan variasi kecepatan q dan ketinggian ρ

$$\min w_1 \sum_{f \in F} \left(\frac{c_f^{q^+} q_f^+}{\bar{v}_f - \underline{v}_f} + \frac{c_f^{q^-} q_f^-}{\bar{v}_f - \underline{v}_f} \right) + w_2 \sum_{f \in F} c_f^j \rho_f$$

$c_f^{q^+}, c_f^{q^-}$: *Cost* untuk positif dan negatif variasi perubahan kecepatan untuk setiap pesawat $(1, \dots, n)$.

c_f^j : *Cost* untuk jumlah perubahan level ketinggian pada setiap pesawat $(1, \dots, n)$.

w_n : Nilai berat untuk setiap *objective function* dengan nilai antara 0 dan 1.

- Variasi kecepatan tidak boleh mengubah kecepatan menjadi lebih besar dari kecepatan maximum dan kecepatan minimumnya

$$(1) \underline{v}_f \leq v_f + q_f \leq \bar{v}_f$$

- q_f^+ dan q_f^- adalah nilai non-negatif dari q .

$$(2) q_f = q_f^+ - q_f^-$$

- Setiap pesawat harus berada di salah satu level ketinggian

$$(3) \sum_{f \in Z^f} v_f^z = 1$$

V_f^z : BINARY, variabel keputusan yang bernilai 1 jika pesawat f terbang pada level ketinggian z dan 0 untuk sebaliknya. Dilakukan untuk setiap pesawat $(1, \dots, \bar{f})$ dan pada setiap level ketinggian $(1, \dots, z)$.

- Perubahan level ketinggian harus seminimal mungkin untuk melihat banyaknya perubahan level yang terjadi

$$(4) \sum_{f \in Z^f} z v_f^z - z_f \leq \rho_f$$

$$(5) z_f - \sum_{f \in Z^f} z v_f^z \leq \rho_f$$

- Melihat ketinggian setelah optimasi

$$(6) \sum_{f \in Z^f} z v_f^z = z'_f$$

z : level ketinggian (1... z)

z'_f : INTEGER, variabel keputusan yang mengambil nilai level ketinggian untuk setiap pesawat (1,..., f).

- ◆ Batasan 7-15 merupakan batasan yang berfungsi untuk pencarian kecepatan optimal.

$$\forall i, j \in F | i < j, \forall z \in Z^i \cap Z^j \wedge f_{c_{ij}} = 0$$

$$(7) (v_i + q_i)(\cos(m_i^*) (1 - pc_{ij}) - \sin(m_i^*) pc_{ij}) - (v_j + q_j)(\cos(m_j^*) (1 - pc_{ij}) - \sin(m_j^*) pc_{ij}) \leq (\bar{v}_i + \bar{v}_j)(1 - \delta_{ijz}^1)$$

$$(8) -(v_i + q_i)(h_i(1 - pc_{ij}) + h'_i pc_{ij}) + (v_j + q_j)(h_j(1 - pc_{ij}) + h'_j pc_{ij}) \leq ((\bar{v}_i |h_i| + \bar{v}_j |h_j|)(1 - pc_{ij}) + (\bar{v}_i |h'_i| +$$

- ◆ Batasan 16-18 merupakan batasan yang berfungsi untuk mencari ketinggian optimal.

$$(16) \delta_{ijz}^5 = 1 \quad \text{if } hth_{ij} + sc_{ij} \geq 1$$

$$(17) v_i^z + v_j^z \geq \delta_{ijz}^5 - 1$$

$$(18) v_i^z + v_j^z \leq 2 - \delta_{ijz}^5$$

hth_{ij} : BINARY PARAMETER, bernilai 1 jika pesawat i dan pesawat j terbang menuju satu sama lain (*head to head*), 0 untuk sebaliknya. ($i, j \in F$).

sc_{ij} : BINARY PARAMETER, bernilai 1 jika pesawat i dan pesawat j memiliki jarak kurang dari *safety distance*, 0 untuk sebaliknya. ($i, j \in F$).

- ◆ Deteksi dilakukan dengan cara melihat letak quadran pesawat j dari sudut pandang pesawat i dan memodifikasi parameter sudut ω .
- ◆ jika berada pesawat j berada :
 - quadran I atau IV maka sudut ω bernilai tetap.
 - quadran II, sudut ω ditambah π
 - quadran III, sudut ω dikurang π
- ◆ Hasil modifikasi sudut ω disimpan dalam parameter sudut $\hat{\omega}$
- ◆ Deteksi *head to head*:

$$hth_{ij} = \begin{cases} 1, & \text{if } \hat{g}_{ij} \leq m_i^* \leq \hat{l}_{ij} \wedge \hat{g}_{ji} \leq m_j^* \leq \hat{l}_{ji} \\ 0 & \end{cases}$$

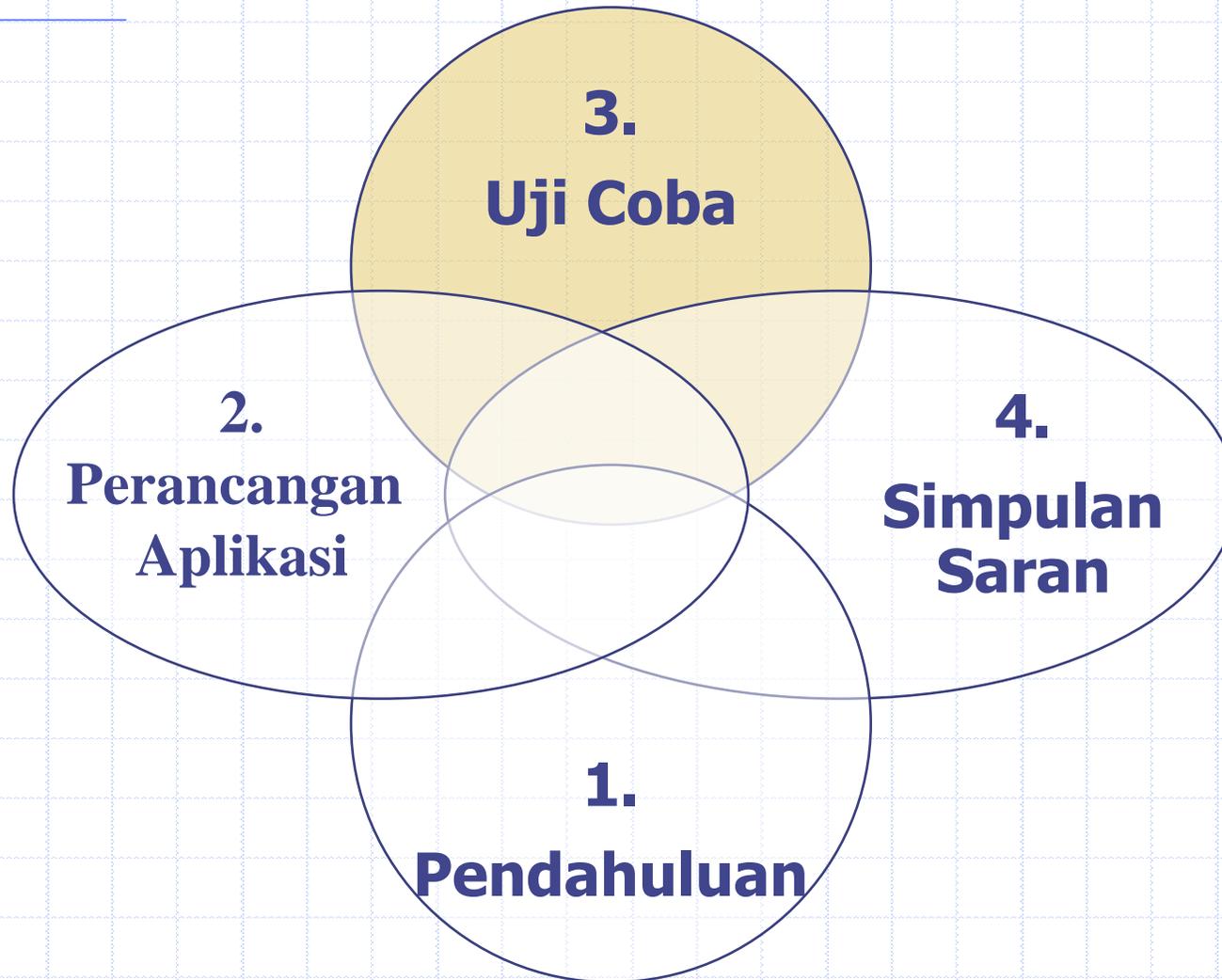
dengan

$$\begin{aligned} \hat{g}_{ij} &= \hat{\omega}_{ij} - \alpha_{ij} \\ \hat{l}_{ij} &= \hat{\omega}_{ij} + \alpha_{ij} \\ \omega_{ij} &= \arcsin\left(\frac{y_i - y_j}{x_i - x_j}\right) \\ \alpha_{ij} &= \arcsin\left(\frac{r_i + r_j}{d_{ij}}\right) \end{aligned}$$

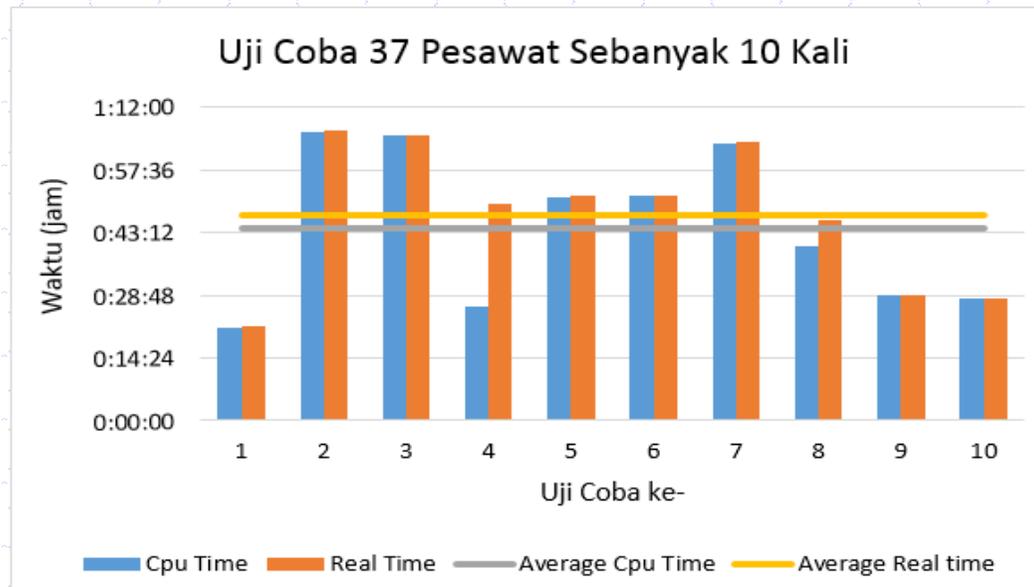
SIMILAR COORDINATES

- ◆ Mendeteksi jika ada dua pesawat yang memiliki jarak kurang dari *safety distance*
- ◆ Deteksi *similar coordinates*:

$$sc_{ij} = \begin{cases} 1, & \text{if } d_{ij} \leq r_i + r_j \\ 0 & \end{cases}$$

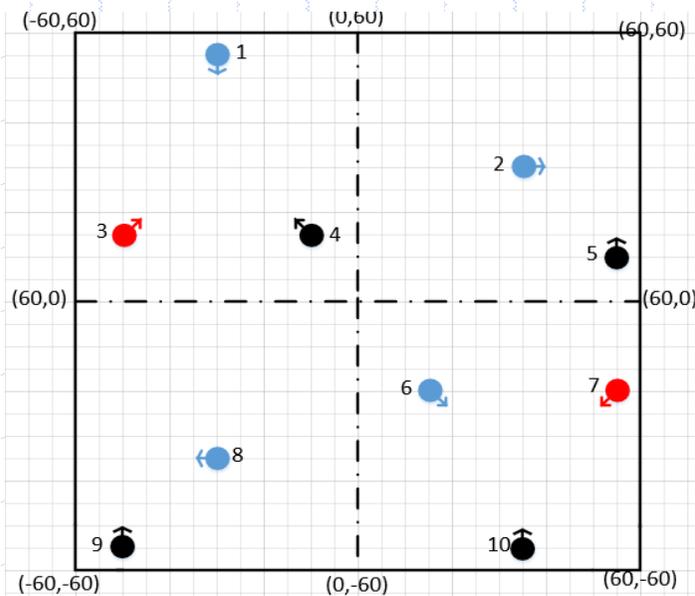


- Uji coba kinerja dilakukan menggunakan data uji 37 pesawat sebanyak 10

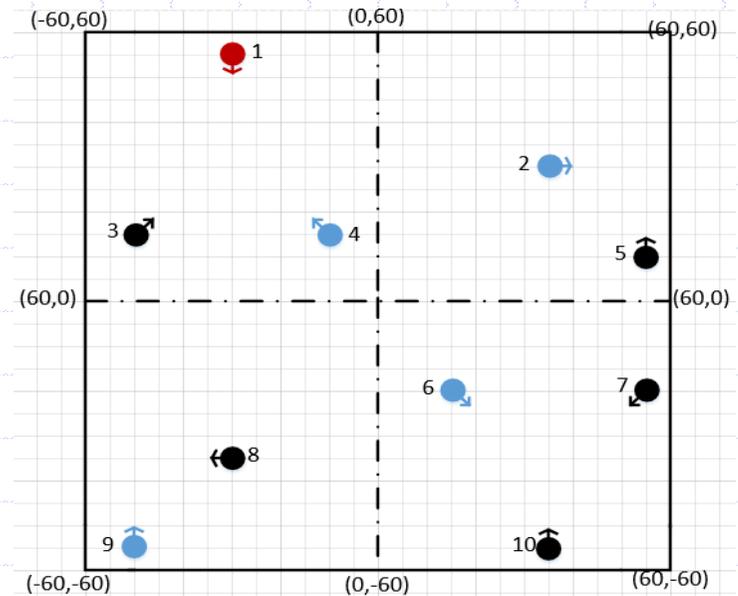


- CPU time* yang dibutuhkan program minimum 21 menit 19 detik, maksimum 1 jam 6 menit 25 detik dan rata-rata 44 menit 17 detik.
- Real time* yang dibutuhkan program minimum 21 menit 46 detik, maksimum 1 jam 6 menit 35 detik dan rata-rata 47 menit 22 detik.

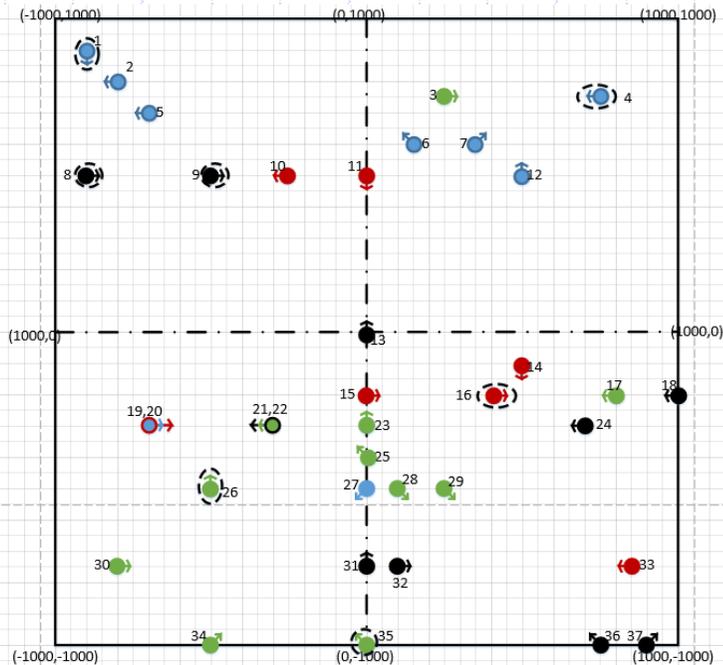
Alonso



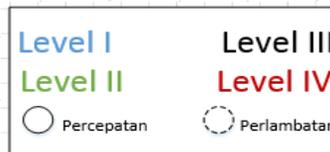
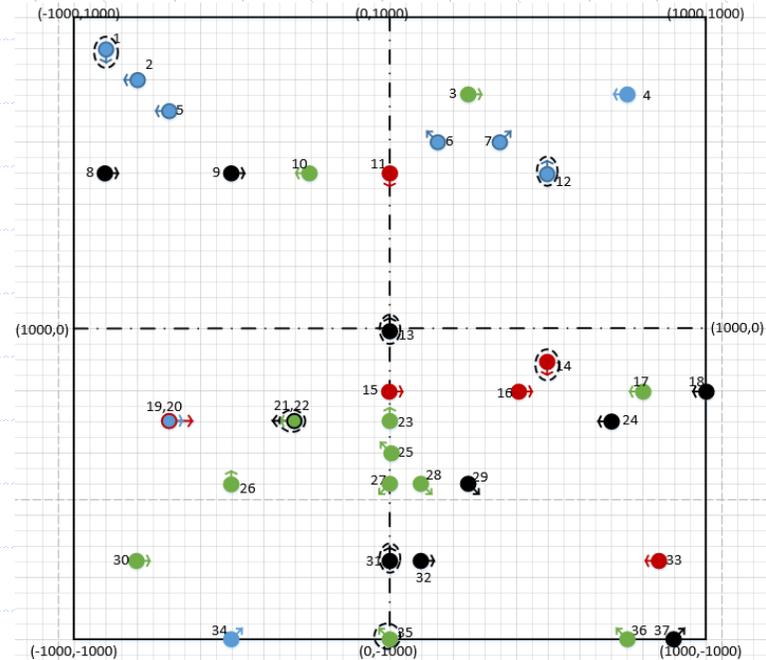
Penulis



ALONSO

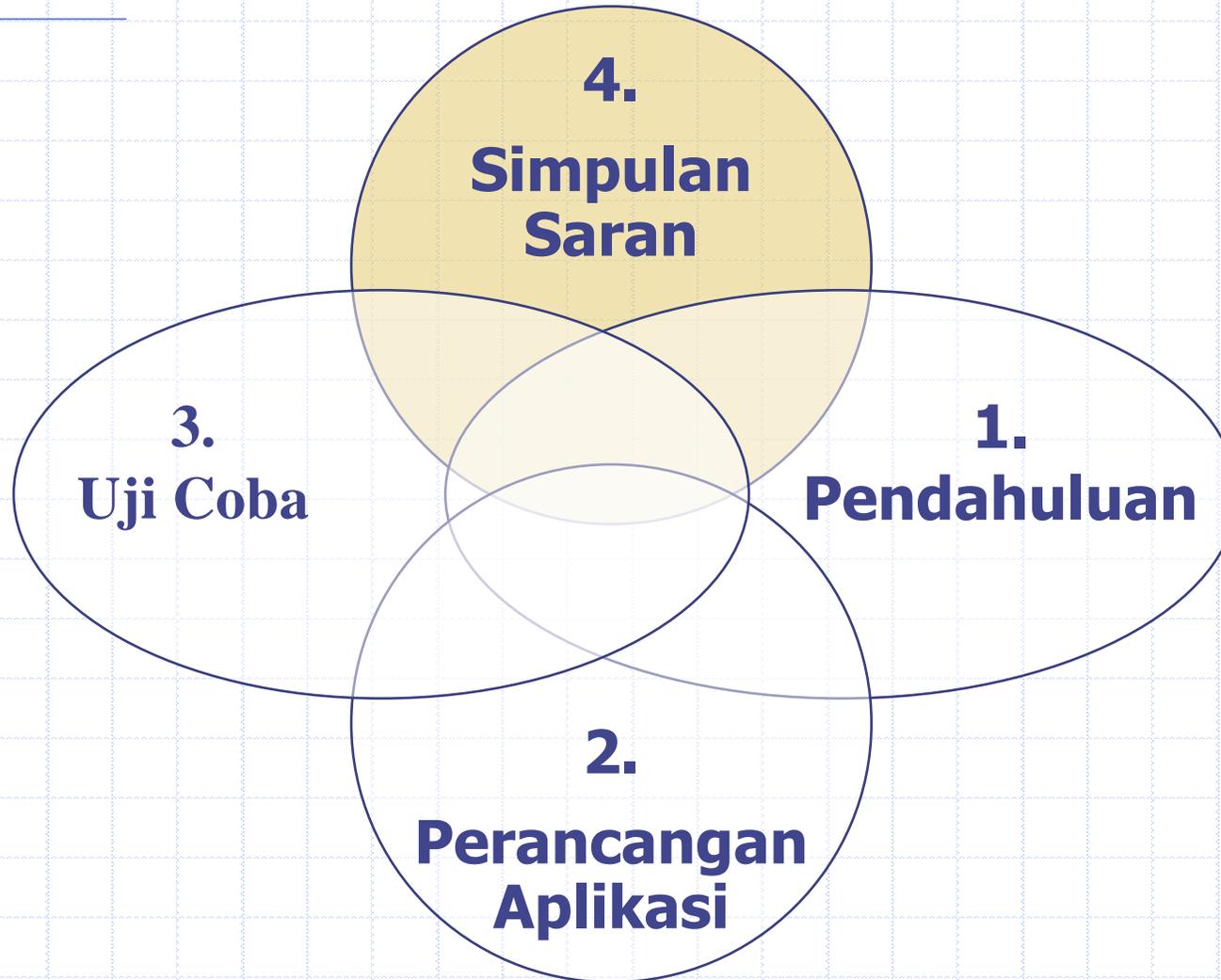


PENULIS



nilai obyektif antara penulis dan Alonso menghasilkan nilai yang sama yaitu 6.5562

SIMPULAN DAN SARAN



- ◆ Model VAC mampu menyelesaikan permasalahan penghindaran konflik karena menggunakan konstruksi geometri untuk mengatasi berbagai situasi konflik.
- ◆ Model VAC dapat diimplementasikan kaku bantu SAS 9.2 yang meliputi modul praproses dan modul optimasi dengan *solver Mixed Integer Linear Programming*.
- ◆ Berdasarkan hasil uji coba, untuk uji kebenaran dengan 10 pesawat yang dilakukan oleh penulis memberikan hasil sama benarnya dengan hasil uji coba yang dilakukan oleh Alonso et al [1] dikarenakan semua konflik yang terjadi pada sebelum optimasi telah dapat dihindari. Sementara, hasil uji kinerja 37 pesawat menunjukkan nilai obyektif 6.5562 dengan jumlah *constraint* sebanyak 29130, 27602 variabel *binary*, dan 37 variabel *continue*. Dengan *cpu time* dan *real time* adalah 44:16.8 menit dan 47:22.2 menit.

- ◆ Diperlukan pengembangan model VAC terkait dengan pengembalian posisi pesawat sesuai *time flight*, sehingga pesawat dapat tiba ditujuan dengan tepat waktu.
- ◆ Penambahan perhitungan sudut α untuk pesawat yang memiliki *safety distance* yang berbeda dengan pesawat lain.
- ◆ Pengembangan model VAC dengan mempertimbangkan faktor luar seperti cuaca, penggunaan bahan bakar, kecepatan angin, dll.
- ◆ Pengembangan model dengan penambahan variabel waktu dan penambahan manuver sudut arah terbang untuk mendapatkan hasil yang optimal dan media penghindaran konflik yang lengkap.
- ◆ Penggunaan wilayah radar ATC sebagai pengganti sektor.



TERIMA KASIH