

## BAB II

### TINJAUAN PUSTAKA



#### 2.1 Jaringan Penukar Kalor (HEN)

Jaringan penukar kalor atau *Heat Exchanger Network (HEN)* didefinisikan sebagai penentuan harga efektif jaringan untuk penukar kalor diantara satu set proses aliran, dimana beberapa pemanasan dan pendinginan yang tidak terpenuhi dengan menukarkan panas diantara aliran-aliran proses tersebut yang harus disediakan oleh utilitas dari luar (misalnya: uap, minyak panas, air pendingin, dan refrigerant). Untuk membangun sebuah jaringan yang aman, fleksibel, dapat beroperasi dengan baik, dan terkontrol maka harus dilakukan perhitungan yang tepat. Berikut informasi yang harus tersedia:

- Kumpulan dari aliran panas yang didinginkan dan aliran dingin yang dipanaskan
- Temperatur masuk dan keluar (*inlet & outlet*) dari semua aliran
- *Specific heats*( $C_p$ ) dan *heat transfer coefficients* ( $h$ ) dari aliran, dengan temperatur tetapnya
- Penurunan tekanan yang diizinkan/diperbolehkan pada aliran
- Ketersediaan utilitas (dengan Temperatur masuk, temperatur keluar, *specific heat*( $C_p$ ),  $h$ , dan biaya)
- Peraturan biaya untuk area penukar panas, dengan pembangunan yang berjalan dan bunga yang berjalan

Dalam pembangkit listrik tenaga uap, dapat diamati dari siklus termodinamika bahwa sintesis dari HEN optimal dengan minimalisasi utilitas, dapat meningkatkan efisiensi siklus. Jadi desain yang berkaitan dengan HEN untuk efisiensi optimal adalah tujuan utama penggunaan teknologi *pinch* dalam optimasi *power plant*[1].

Teknologi *pinch* adalah salah satu alat untuk menyelesaikan masalah diatas. Solusi tersebut merekomendasikan kebutuhan dari keadaan target, misalnya memprediksi performan apa yang paling baik yang dapat memungkinkan tercapainya suatu proses, sebelum percobaan tersebut benar-benar tercapai. Dengan demikian, penargetan mengizinkan proses mekanik untuk menentukan utilitas minimum yang dibutuhkan, area, unit, *shells*, dan kebutuhan biaya yang aktual untuk perancangan dari jaringan penukar kalor)[2].

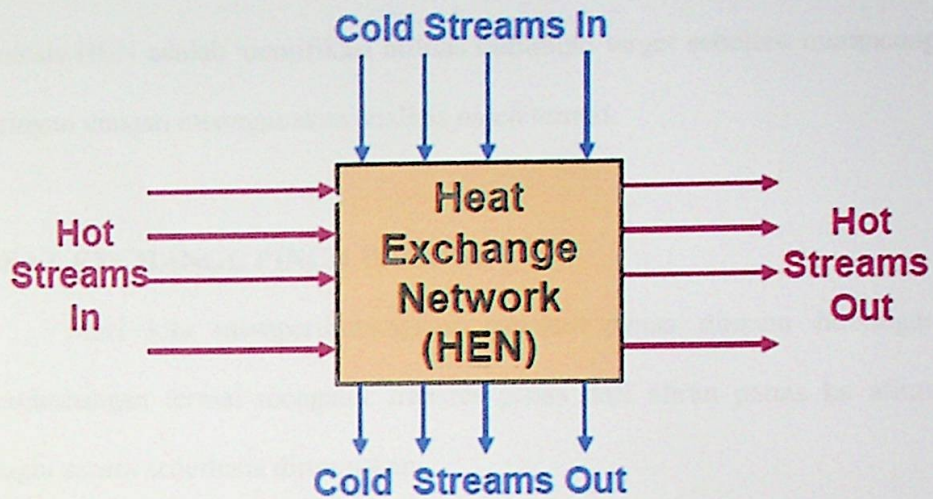
Analisis *Pinch* adalah metodologi untuk meminimalkan konsumsi energi dari proses kimia dengan menghitung target energi termodinamika layak (atau konsumsi energi minimum) dan mencapainya dengan mengoptimalkan *heat recovery systems*, metode pasokan energi dan kondisi operasi proses. Hal ini juga dikenal sebagai integrasi proses, integrasi panas, integrasi energi atau teknologi *pinch*[3].

Dalam proses yang khas, biasanya ada beberapa aliran panas yang harus didinginkan dan beberapa aliran dingin yang harus dipanaskan. Penggunaan pendingin eksternal dan utilitas pemanasan (misalnya, air pendingin, pendingin, uap, minyak pemanas, dll) untuk mengatasi semua

pemanasan dan pendinginan yang wajib merupakan biaya yang tidak efektif. Memang, integrasi pemanasan dan pendinginan yang wajib dapat menyebabkan biaya yang signifikan menurun. Konsep utama adalah mentransfer panas dari proses aliran panas ke proses aliran dingin sebelum utilitas eksternal digunakan. Hasil integrasi dari panas ini adalah pengurangan simultan pemanasan dan pendinginan yang wajib dari utilitas eksternal.

Masalah sintesis HEN dapat dinyatakan sebagai berikut:

Mengingat sejumlah  $N_H$  proses aliran panas (untuk didinginkan) dan  $N_C$  sejumlah proses aliran dingin (untuk dipanaskan), didinginkan untuk mensintesis jaringan yang hemat biaya alat penukar panas yang dapat mentransfer panas dari aliran panas ke aliran dingin. Mengingat juga kapasitas panas (debit spesifik  $x$  panas) dari setiap proses aliran panas,  $FC_{P,u}$ ; menyuplai (inlet) suhu,  $T_u^s$ , dan target (outlet) suhu,  $T_u^t$ ; di mana  $u = 1, 2, \dots, N_H$ . Selain itu, kapasitas panas,  $FC_{P,v}$ , suplai dan Target suhu,  $t_v^s$  dan  $t_v^t$ , diberikan untuk setiap proses aliran dingin, di mana  $v = 1, 2, \dots, N_C$ . Ketersediaan pelayanan  $N_{HU}$  utilitas pemanasan dan  $N_{CU}$  utilitas pendinginan yang suplai dan suhu Target (tapi tidak mengalir) diketahui. Gambar 2.1 adalah representasi skematis dari pernyataan masalah HEN.



Gambar 2.1 Sintesis dari HEN

(sumber: [4])

Untuk sistem yang diberikan, sintesis HEN mensyaratkan jawaban beberapa pertanyaan:

- Utilitas pemanasan/pendinginan mana yang harus digunakan?
- Berapa beban panas yang optimal untuk dibuang / ditambahkan oleh masing-masing utilitas?
- Bagaimana seharusnya aliran panas dan dingin dicocokkan ( pasangan aliran)?
- Sistem konfigurasi apa yang optimal (misalnya, bagaimana seharusnya penukar panas diatur? Apakah ada pemisahan aliran dan pencampuran?)?

Banyak metode yang telah dikembangkan untuk sintesis HEN. Metode ini telah ditinjau oleh Shenoy (1995), Linnhoff (1993), Gundersen



dan Naess (1988) dan Douglas (1988). Salah satu kemajuan penting dalam sintesis HEN adalah identifikasi utilitas minimum target sebelum merancang jaringan dengan menggunakan analisis *pinch* termal.

#### 4.2. HEAT-EXCHANGE PINCH DIAGRAM

Mari kita mempertimbangkan penukar panas dimana hubungan keseimbangan termal mengatur transfer panas dari aliran panas ke aliran dingin secara sederhana dirumuskan

$$T = t \quad (1)$$

Dengan menggunakan minimal pertukaran panas  $\Delta T_{min}$ , salah satu dapat membangun satu-ke-satuan korespondensi antara aliran suhu panas dan dingin dimana perpindahan panasnya layak, yaitu

$$T = t + \Delta T_{min} \quad (2)$$

Persamaan ini memastikan bahwa pertimbangan perpindahan panas dari hukum kedua termodinamika terpenuhi. Untuk pasangan suhu tertentu yang diberikan  $(T, t)$  secara hukum termodinamika dan prakteknya layak untuk mentransfer panas dari setiap aliran panas yang suhunya lebih besar atau sama dengan  $T$  ke setiap aliran dingin yang suhunya lebih kurang atau sama dengan  $t$ . Kesetimbangan termal adalah kasus spesial dari kesetimbangan pertukaran massa dengan  $T, t$ , dan  $\Delta T_{min}$ , sesuai dengan  $y_i, x_j$  dan  $\varepsilon_j$ , masing-masing, sedangkan nilai  $m_j$  dan  $b_j$  adalah satu dan nol, masing-masing. Tabel 2.1 merangkum istilah analog di MENs dan HENs.

Tabel 2.1 MENs vs HENs

(sumber: [4])

| MENs                          | HENs                            |
|-------------------------------|---------------------------------|
| Transferred commodity: Mass   | Transferred commodity: Heat     |
| Donors: Rich streams          | Donors: Hot streams             |
| Recipient: Lean streams       | Recipient: Cold streams         |
| Rich composition: $y$         | Hot temperature: $T$            |
| Lean composition: $x$         | Cold temperature: $t$           |
| Slope of equilibrium: $m$     | Slope of equilibrium: 1         |
| Intercept of equilibrium: $b$ | Intercept of equilibrium: 0     |
| Driving force: $\varepsilon$  | Driving force: $\Delta T^{min}$ |

Untuk mencapai penggunaan minimum utilitas pemanasan dan pendinginan, maka perlu memaksimalkan pertukaran panas antara aliran-aliran proses. Dalam konteks ini, kita dapat menggunakan teknik grafis yang sangat berguna yang disebut sebagai "*termal-pinch diagram*." Teknik ini terutama didasarkan pada karya Linnhoff dan rekan kerjanya (misalnya Linnhoff dan Hindmarsh, 1983), Umeda et al. (1979), dan Hohmann (1971). Langkah pertama dalam membangun *termal-pinch diagram* adalah dengan menciptakan global representasi untuk semua aliran panas dengan memplot entalpi yang dipertukarkan oleh setiap proses aliran panas dibandingkan dengan temperaturnya. Oleh karena itu, aliran panas kehilangan *sensible heat* yang direpresentasikan sebagai panah dimana ekor panah untuk suplai suhu dan kepalanya mengarah pada suhu target. Dengan asumsi *heat capacity* konstan selama rentang operasi, kemiringan setiap panah adalah sama dengan  $F_u C_{p,u}$ . Jarak vertikal antara ekor dan kepala setiap panah merupakan entalpi yang hilang oleh aliran panas menurut persamaan berikut:

$$\text{Heat lost dari ke-}u \text{ aliran panas } HH_u = F_u C_{p,u} (T_u^s - T_u^t),$$

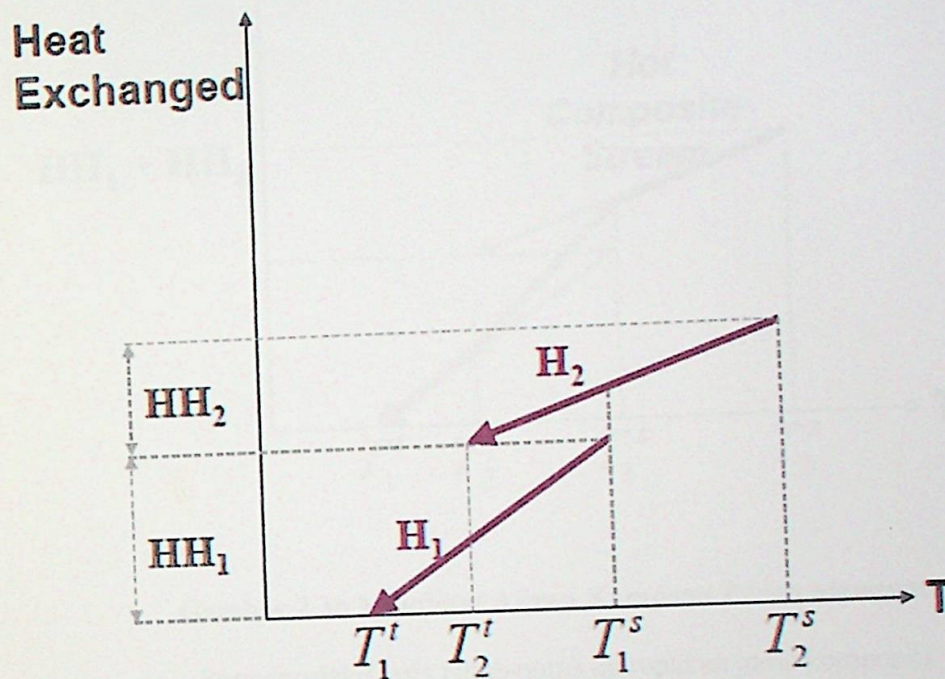
dimana

$$u = 1, 2, \dots, N_H. \quad (3)$$

Catatan bahwa aliran apapun dapat bergerak naik atau turun sambil menjaga jarak vertikal yang sama antara kepala dan ekor panah dan



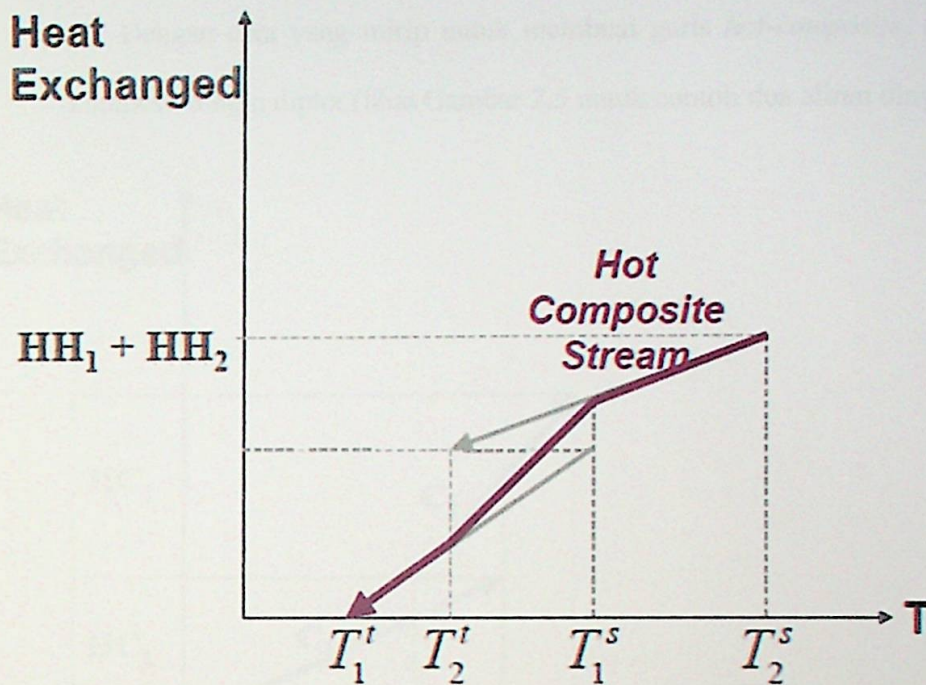
menjaga temperatur suplai dan temperatur target yang sama. Gambar 2.3 mengilustrasikan konsep untuk dua aliran panas.



Gambar 2.3a Konsep Aliran Panas

(sumber: [4])





**Gambar 2.3b Membuat Aliran Komposit Panas Menggunakan Superposisi (Garis putus-putus merupakan garis komposit)**

(sumber: [4])

Selanjutnya, skala temperatur dingin,  $t$ , dibuat dalam satu-ke-satuan korespondensi dengan skala temperatur panas,  $T$ , menggunakan Persamaan. (2). Entalpi setiap aliran dingin diplot terhadap skala temperatur dingin,  $t$ . Jarak vertikal antara kepala panah dan ekor untuk aliran dingin diberikan oleh

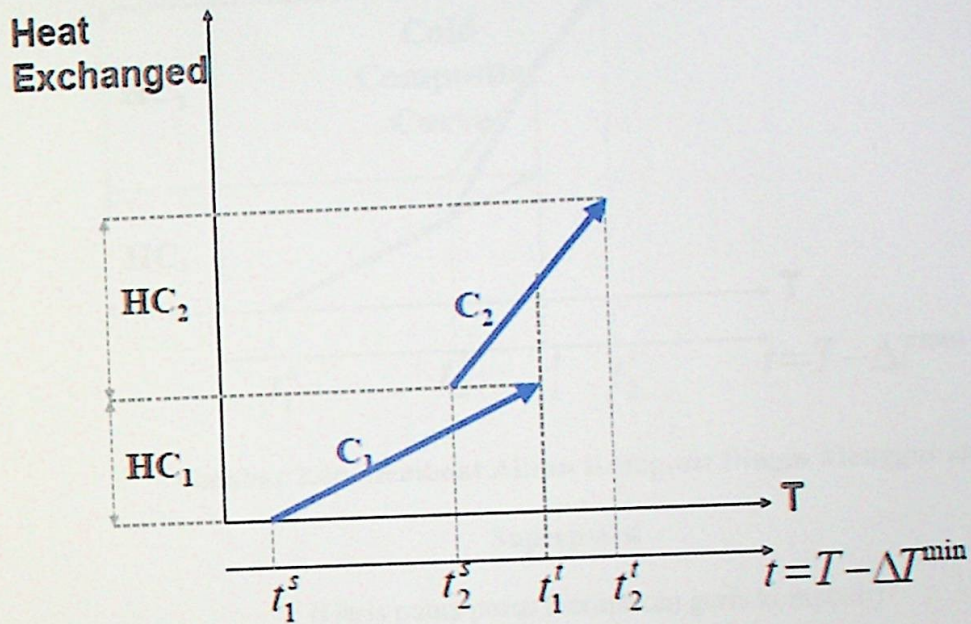
$$\text{Panas yang diperoleh oleh ke-}v\text{ aliran dingin } HC_v = F_v C_{P,v} (T_v^t - T_v^s),$$

di mana



$$v = 1, 2, \dots, N_C. \quad (4)$$

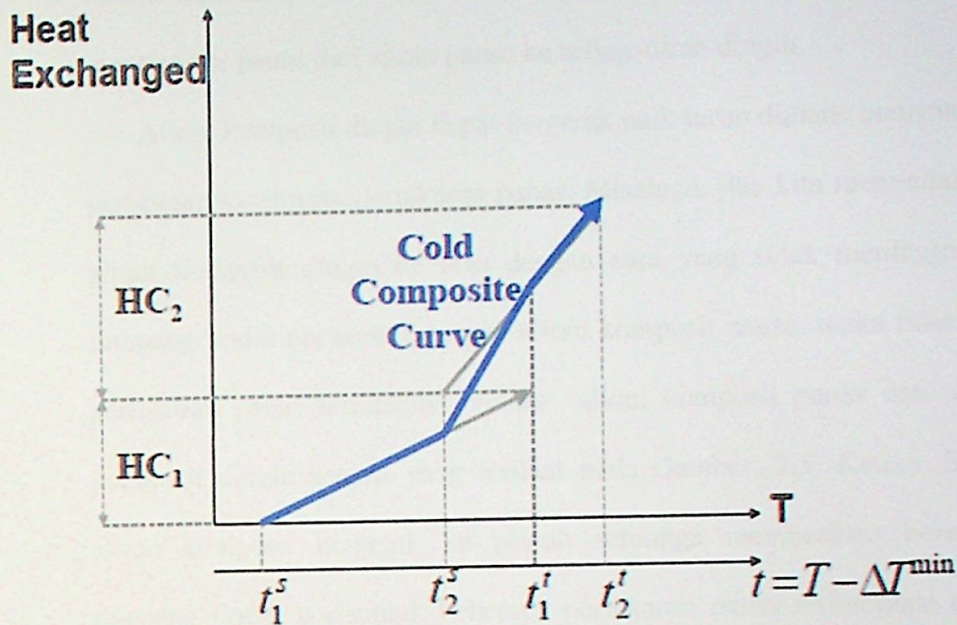
Dengan cara yang mirip untuk membuat garis *hot-composite*, aliran komposit dingin diplot (lihat Gambar 2.5 untuk contoh dua aliran dingin).



Gambar 2.4a Konsep Aliran Dingin

(sumber: [4])





Gambar 2.4b Membuat Aliran Komposit Dingin Menggunakan Superposisi

(Garis putus-putus merupakan garis komposit)

(sumber: [4])

Selanjutnya, kedua aliran komposit diplot pada diagram yang sama (Gambar 2.5). Pada diagram ini, kelayakan termodinamika pertukaran panas dijamin ketika pada setiap tingkat pertukaran panas (yang sesuai dengan garis horizontal), temperatur dari aliran komposit dingin terletak di sebelah kiri dari aliran komposit panas (yaitu, suhu panas lebih besar atau sama dengan dingin suhu ditambah dengan temperatur pendekatan minimum). Oleh karena itu, untuk satu set tertentu sesuai dengan

temperatur, secara hukum termodinamika dan prakteknya layak untuk mentransfer panas dari aliran panas ke setiap aliran dingin.

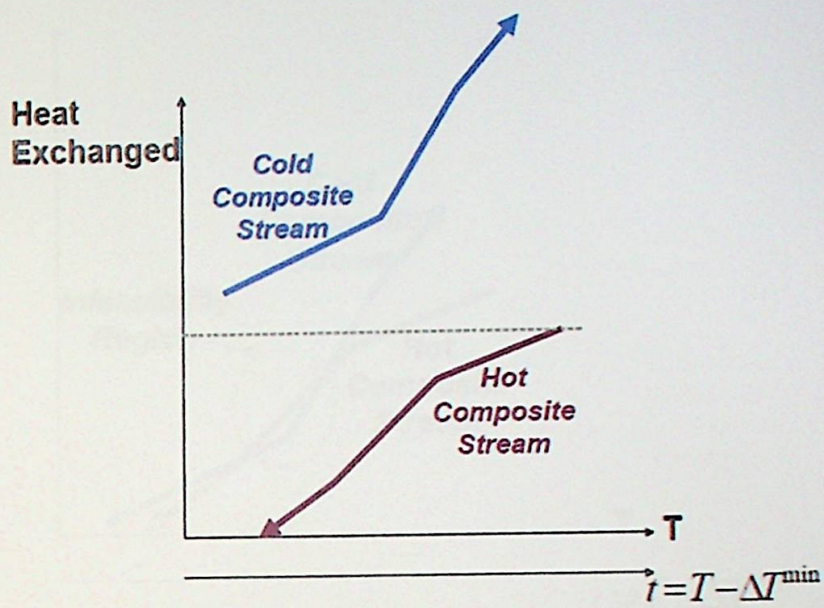
Aliran komposit dingin dapat bergerak naik turun dimana menyatakan perbedaan keputusan pertukaran panas. Misalnya, jika kita memindahkan aliran komposit dingin ke atas dengan cara yang tidak meninggalkan tumpang tindih horisontal dengan aliran komposit panas, maka tidak ada pertukaran panas terintegrasi antara aliran komposit panas dan aliran komposit dingin seperti yang terlihat pada Gambar. 2.5. Ketika dingin aliran komposit bergerak ke bawah sehingga memberikan beberapa tumpang tindih horisontal, beberapa pertukaran panas terintegrasi dapat dicapai (Gambar 2.6). Namun, jika aliran komposit dingin dipindahkan bawah seperti sebagian dari aliran dingin ditempatkan di sebelah kanan aliran komposit panas, sehingga membuat suatu ketidaklayakan (Gambar 2.7). Oleh karena itu, situasi yang optimal dibuat ketika aliran komposit dingin mengarah vertikal sampai menyentuh aliran komposit yang banyak campuran sambil membentang sepenuhnya di sebelah kiri aliran komposit panas pada setiap tingkat horisontal. Oleh karena itu, aliran komposit dingin dapat mengarah turun sampai menyentuh aliran komposit panas. Titik dimana dua komposit aliran sentuh disebut "*thermal pinch point*." Seperti Gambar. 2.8. menunjukkan, salah satunya dapat menggunakan diagram *pinch* untuk menentukan pemanasan minimum dan kebutuhan utilitas pendinginan. Sekali lagi, garis komposit dingin tidak bisa mengarah turun lebih jauh, jika tidak, bagian dari aliran komposit dingin



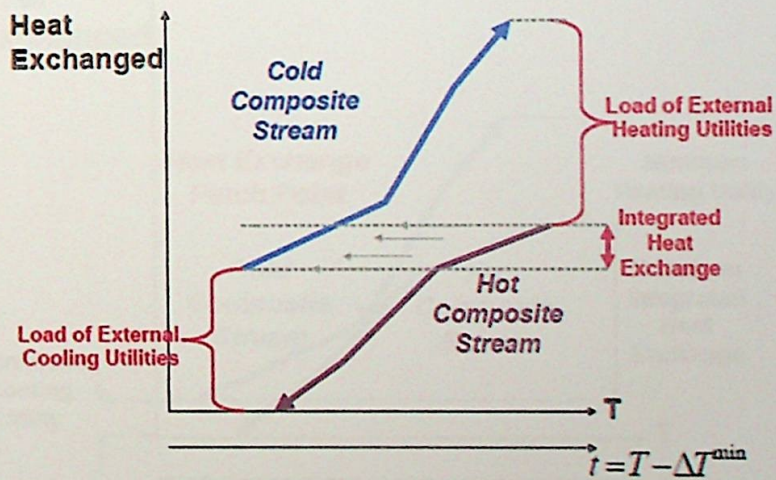
akan berada disebelah kanan aliran komposit panas, menyebabkan termodinamika tidak berlaku. Di sisi lain, jika menaikan aliran komposit dingin (yaitu, melewati panas sepanjang *pinch*), kemungkinan integrasi panas berkurang, dan akibatnya, penambahan utilitas pemanasan dan pendinginan diperlukan. Oleh karena itu, untuk meminimumkan penggunaan utilitas, aturan desain berikut harus diperhatikan:

- Tidak ada panas yang boleh melewati sepanjang *pinch*
- Di atas *pinch*, tidak boleh ada utilitas pendinginan yang digunakan
- Di bawah *pinch*, tidak boleh ada utilitas pemanasan yang digunakan

Aturan pertama diilustrasikan oleh Gambar. 2.9. Bagian dari aliran panas melalui *pinch* ( $\alpha$ ) menghasilkan *double penalty*: peningkatan  $\alpha$  di kedua utilitas pemanasan dan pendinginan. Aturan kedua dan ketiga dapat dijelaskan dengan mencatat bahwa tidak ada surplus kapasitas pendinginan di atas *pinch*. Penambahan utilitas pendinginan di atas *pinch* akan mengganti beban yang dapat dihapus (hampir tanpa biaya operasi) oleh proses aliran dingin. Argumen yang sama dapat dilakukan terhadap penggunaan utilitas pemanas di bawah *pinch*.

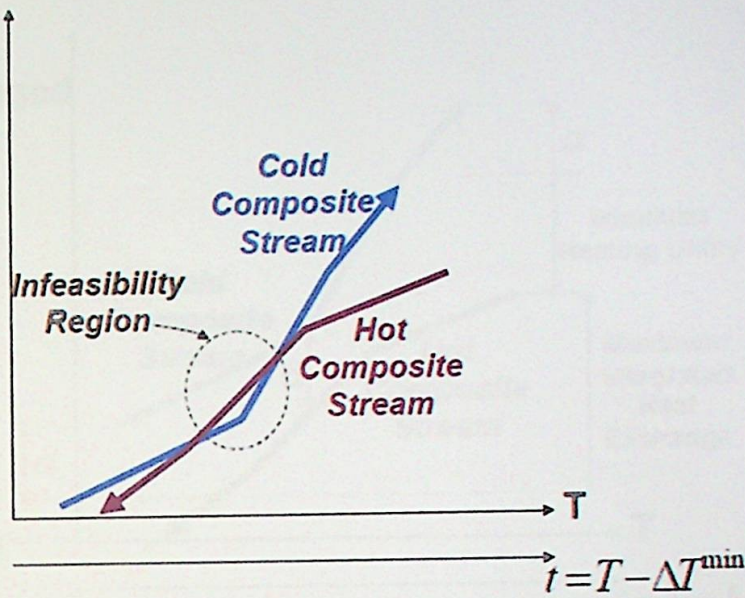


Gambar 2.5 Penempatan Aliran Komposit dengan tidak ada Integrasi Panas  
(sumber: [4])

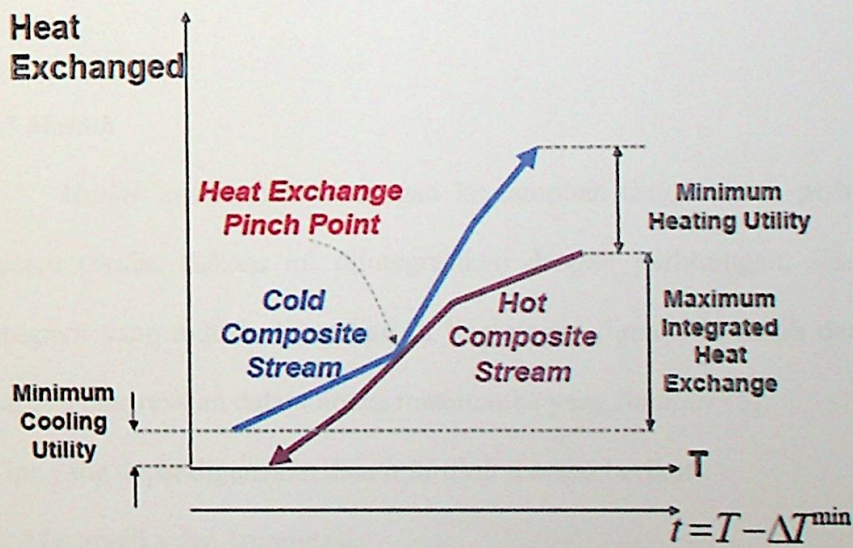


Gambar 2.6 Sebagian Integrasi Panas  
(sumber: [4])



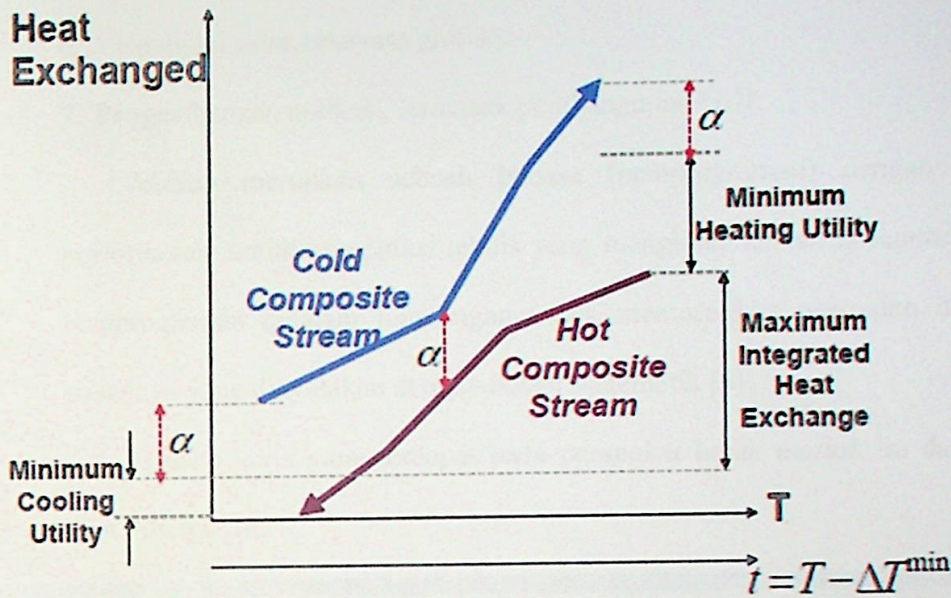


Gambar 2.7 Integrasi Panas yang tidak Layak  
(sumber: [4])



Gambar 2.8 Thermal Pinch Diagram  
(sumber: [4])





Gambar 2.9 Aturan Terkait dengan Panas yang melalui Pinch  
(sumber: [4])

#### 4.3 Matlab

*Matlab* adalah bahasa dengan kemampuan tinggi untuk perhitungan secara teknik. Bahasa ini diintegrasikan dengan perhitungan, visualisasi, program yang mudah digunakan di lingkungan dimana masalah dan solusi dapat diekspresikan dalam notasi matematika yang *familiar* [5].

Tipe yang dapat digunakan dalam *Matlab* sebagai berikut:

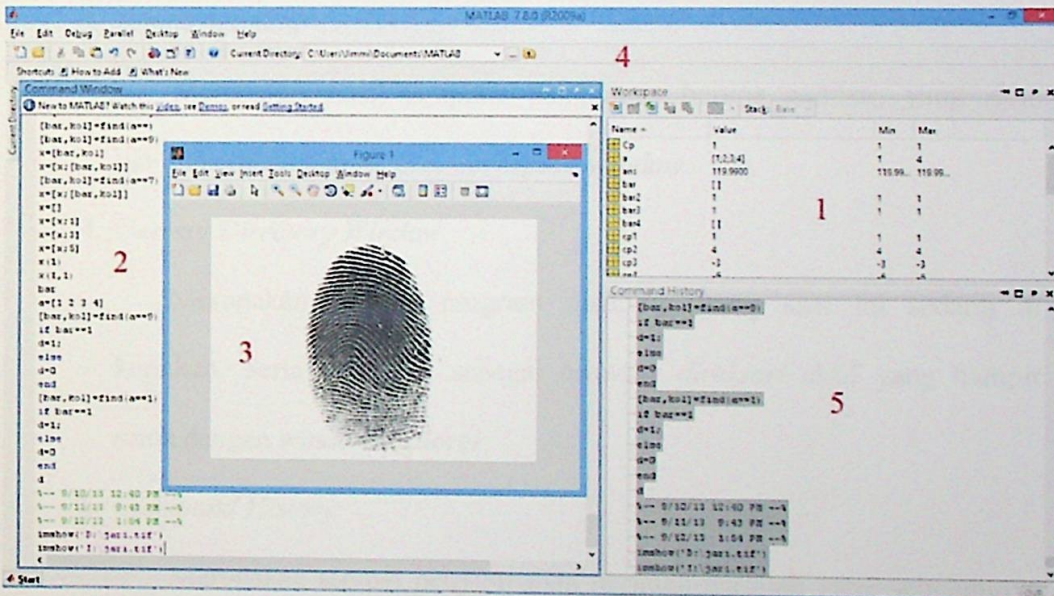
1. Matematika dan komputasi.
2. Pengembangan algoritma.
3. Akuisisi data.
4. *Modeling*, simulasi dan *Prototyping*.



5. Analisis data, penjelajahan dan visualisasi.
6. Visualisasi sains rekayasa grafis.
7. Pengembangan aplikasi, termasuk pembangunan GUI.

*Matlab* merupakan sebuah bahasa (pemrograman) dengan *high-performance* untuk komputasi teknis yang mengintegrasikan komputasi dan pemrograman didalam lingkungan untuk memecahkan persoalan dengan solusinya yang dinyatakan dengan notasi matematik [6].

Ruang kerja yang terdapat pada perangkat lunak *matlab* ini dapat di lihat dibawah ini :



Gambar 2.10 Ruang Kerja *Matlab*

### 1. *Workspace Browser*

Merupakan jendela yang akan menampilkan variabel-variabel yang dijalankan atau pernah dibuat. Berfungsi sebagai navigator bagi pemakai dalam penyediaan informasi mengenai variabel yang sedang aktif pada saat pemakaian.

### 2. *Command Window*

Merupakan jendela untuk pembuatan program yang dimana pembuatan logika program. Pembuatan program ini jika dibuat di dalam *command window* akan mengeksekusi program perbaris jika ditekan *enter*.

### 3. *Figure Window*

Merupakan hasil tampilan pengerjaan berupa gambar, yang akan tampil berbeda tempat dari *workspace window*.

### 4. *Current Directory Window*

Merupakan dimana program disimpan yang saat ini sedang di kerjakan. Serta berfungsi sebagai *browser direktori* aktif yang hampir sama dengan *window explorer*.

### 5. *Command History*

Merupakan tempat penyimpanan perintah-perintah yang sebelumnya telah dieksekusi atau dijalankan.

Untuk pengeksekusian sintaks pemrograman secara banyak baris yang akan dieksekusi, dapat menggunakan *m-file*. Pada *Matlab editor* ini memiliki kemampuan dalam mendeteksi kesalahan pengetikan sintaks oleh pemrogram. Bentuk dari *m-file* pada *matlab* dapat dilihat dibawah ini :



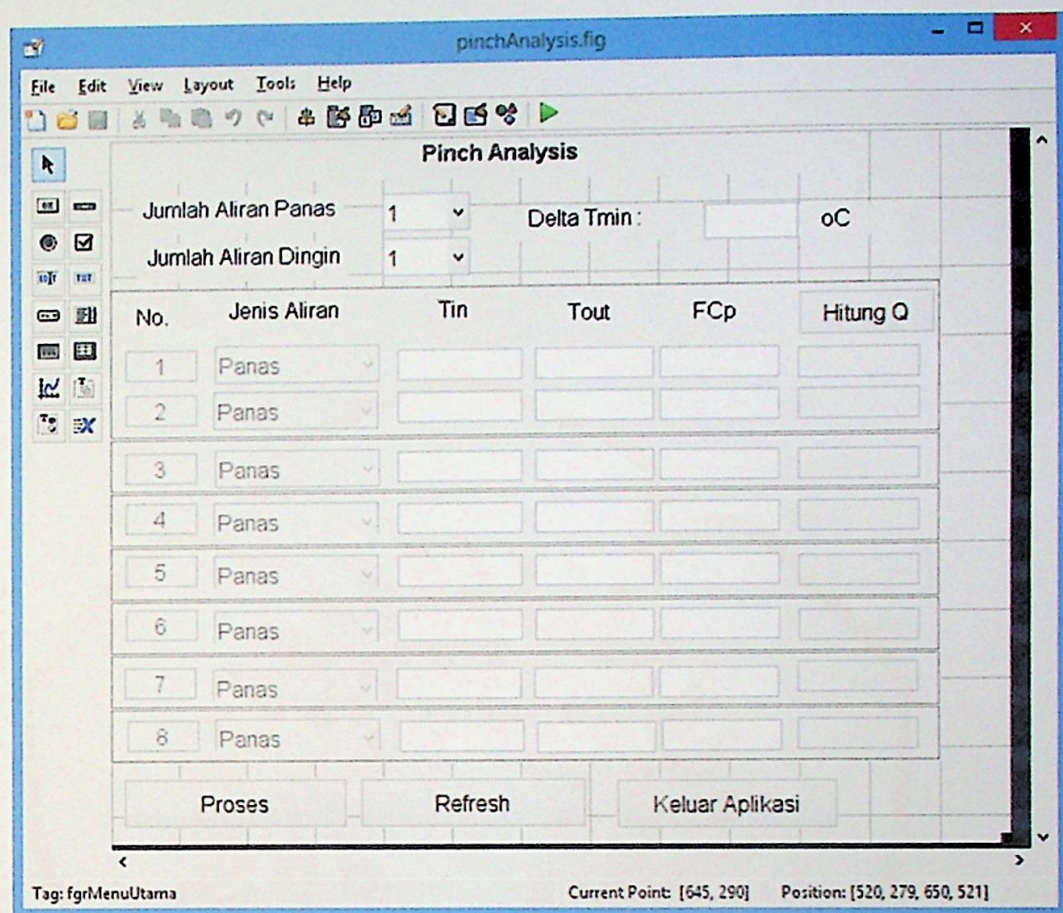
```

Editor - 1\Matlab\script\UASR1\Jurnal\Lang\trajektor\fp\Proses.m
File Edit View Go Cell Tools Debug Desktop Window Help
- 12 + + 11 x
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
88 - end
89 - %
90 - t=[];
91 - a=axis(t1,t2);
92 - b=axis(t1,t2);
93 - for t=0:1:a
94 -     t=[t;2];
95 - end
96 - if(a==t1 && a==b) aliran_panas
97 -     plot(handles.a, b, 'r', 'LineWidth', 2); %garis aliran panas
98 -     plot(handles.a, b, 'v', 'LineWidth', 2); %arah anak panah aliran panas
99 -     a=[t1;t2];
100 -     w=[w1;t2];
101 -     t1=[t2;0.01;t2];
102 -     cp2=double(cp2);
103 - else if (a==t1 && a==b) aliran_dingin
104 -     plot(handles.a, b, 'b', 'LineWidth', 2); %garis aliran dingin
105 -     plot(handles.a, b, 'w', 'LineWidth', 2); %arah anak panah aliran dingin
106 -     a=[t1+deltaTmin;t2+deltaTmin];
107 -     w1=[t1+deltaTmin;t2+deltaTmin];
108 -     t1=[t1+deltaTmin;0.01;t2+deltaTmin];
109 -     cp2=double(cp2);
110 - end
111 -
112 - t=[];
113 - a=axis(t1,t2);
114 - b=axis(t1,t2);
115 - for t=0:1:a
116 -     t=[t;2];
117 - end
118 - if(a==t1 && a==b)
119 -     plot(handles.a, b, 'r', 'LineWidth', 2);

```

Gambar 2.11 Tampilan M-File pada Matlab





Gambar 2.12 Tampilan GUI pada *Matlab*