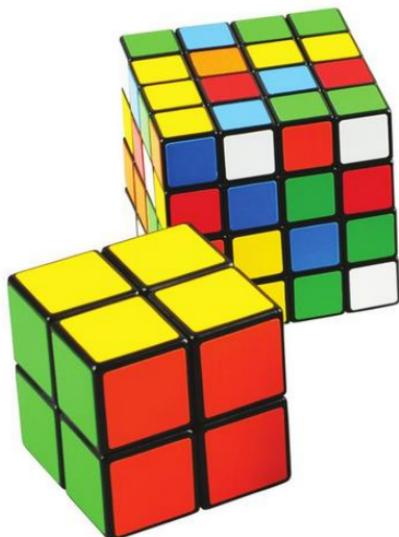


Langkah Mudah Menjadi  
*Master*  
**Rubik**



Abel Brata

# Langkah Mudah Menjadi Master Rubik

**Penulis:** Abel Brata

**Editor:** Rizal Khadafi

**Proof Reader:** Dewi Fita

**Penata Letak:** Yasinta Mutiara Aini

**Desainer Sampul:** Yasinta Mutiara Aini

**Fotografer:** Hadi Iswa

**Ilustrator:** Abel Brata

**Penerbit:** Bukuné

**Redaksi:**

Jln. Haji Montong No. 57 Ciganjur-Jagakarsa

Jakarta Selatan 12630

Telp. (021) 78883030 ext. 228, 207

Faks. (021) 7270996

E-mail: redaksi@bukune.com

Website: www.bukune.com

**Pemasaran:**

**Kawah Media**

Jl. Moh. Kahfi 2 No. 12

Cipedak Jagakarsa Jakarta Selatan

Telp. (021) 7888 1000 ext 120, 121, 122

Faks. (021) 7888 2000

E-mail: kawahmedia@gmail.com

Website: www.kawahdistributor.com

Cetakan pertama, Maret 2010

Hak cipta dilindungi undang-undang

Brata, Abel

*Langkah Mudah Menjadi Master Rubik / 'Abel Brata'* ; penyunting, Rizal

Khadafi – cet.1 – Jakarta: Bukuné, 2010

vi + 90 hlm; 13 x 19 cm

ISBN 602-8066-56-7

I. Hobi

II. Rizal Khadafi

# Daftar Isi

Metode Solving 4x4x4.....	1
Pengantar.....	1
Metode Menyelesaikan 4x4x4.....	2
Menyelesaikan Center.....	2
Menggabungkan Setiap Pasang Double-Edge (Dedge)....	7
Menyelesaikan 4x4x4 Sebagai Sebuah 3x3x3.....	15
Metode Solving 5x5x5.....	19
Pengantar.....	19
Metode Menyelesaikan 5x5x5.....	20
Menyelesaikan Center.....	20
Menggabungkan Triple-Edge (Tredge).....	24
Menyelesaikan 5x5x5 Sebagai Sebuah 3x3x3.....	29
Metode Solving 2x2x2.....	31
Pengantar.....	31
Menyelesaikan 2x2x2 dalam Tiga Langkah Mudah.....	32
Menyelesaikan Layer Pertama.....	32
Menyamakan Warna Sisi Atas.....	33
Menempatkan Corner ke Posisi yang Benar.....	35
Metode Tingkat Lanjut (Ortega Method).....	36
Menyamakan Warna Satu Sisi.....	36
Menyamakan Warna Sisi Sebaliknya.....	37
Menyelesaikan Semuanya.....	37

Metode Solving Pyraminx.....	39	Profil Speedcuber.....	87
Mengenal Pyraminx dan Strukturnya.....	39	Muhammad Irlil Khairul Anam.....	87
Menyelesaikan Pyraminx Layer-By-Layer.....	40	Shotaro "Macky" Makisumi.....	89
Membetulkan Corner Layer Pertama (Base Layer).....	40		
Menyelesaikan Pondasi Piramid.....	41		
Menyelesaikan Puncak Piramid.....	42		
Tip Mempercepat Solving Piramid.....	46		
Metode Blindfold.....	49		
Blindfold Cubing.....	49		
Metode ABC (Abel Brata's Cycle).....	50		
Menyelesaikan Edge.....	51		
Menyelesaikan Corner.....	53		
Parity.....	54		
Teknik Tambahan.....	55		
Algoritma Tambahan.....	60		
Contoh Solving BLD.....	61		
Metode Turbo.....	64		
Blindfold Pyraminx.....	67		
Contoh Solving Pyraminx BLD.....	69		
Memorisasi.....	70		
Visual.....	71		
Alfabet atau Angka.....	71		
PAO (Person-Action-Object).....	73		
Image Based.....	73		
Journey Method.....	73		
Letter Pair.....	74		
Cara Menyelesaikan Cube 3x3x3 dengan Gambar/Logo.....	77		
Pretty Patterns 3x3x3.....	79		

## Metode Solving 4x4x4

### 1. Pengantar

Cube 4x4x4 atau cukup disebut dengan 4x4x4 dikenal sebagai cube “genap” karena dimensi panjang, lebar, dan tingginya adalah 4 yang merupakan angka genap. Hal ini

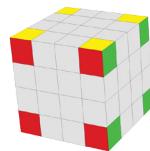
berakibat pada 4x4 tidak terdapat center yang *fixed* atau tetap pada posisinya, sehingga kita tidak dapat menggunakan center sebagai panduan warna-warna pada sisi cube seperti cube berdimensi ganjil (3x3x3 dan 5x5x5).

Jika kamu tidak hafal skema warna 4x4x4, ada cara mudah untuk menentukan warna suatu sisi pada 4x4x4. Yaitu dengan menyelesaikan semua corner terlebih dahulu seperti pada gambar di bawah. Setelah itu kita dapat mengetahui warna dari suatu sisinya.

#### Tahukah kamu?

Jumlah kemungkinan posisi dari sebuah cube 4x4x4 adalah 7,4 quat-tuordecillion ( $7.4 \times 1045$ ). Dan hanya satu di antaranya adalah solusi!

Pada saat buku ini ditulis, rekor dunia untuk Single Solve 4x4 adalah 36,46 detik yang dicetak oleh Dan Cohen pada World Championship 2009 di Jerman. Sedangkan rekor dunia untuk Average dicetak oleh Feliks Zemdegs pada Melbourne Summer Open 2010.

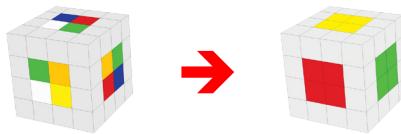


## 2. Metode Menyelesaikan 4x4x4

Metode solving 4x4x4 yang digunakan oleh kebanyakan cuber di dunia sering disebut dengan metode “3x3x3 Reduction”, yaitu membuat bentuk atau susunan 4x4x4 menjadi seperti 3x3x3 kemudian menyelesaiannya dengan metode solving 3x3x3.

Langkah-langkahnya adalah sebagai berikut:

### Menyelesaikan Center



Langkah pertama adalah menyelesaikan semua (24 buah) center piece sehingga menyerupai center pada sebuah 3x3x3. Pastikan susunan dan letak warna-warnanya sesuai dengan skema warna (lihat tip untuk mencari warna sisi di bagian sebelumnya).

#### ◆ Menyelesaikan Center Pertama

Sama seperti membuat cross pada 3x3x3, membuat center pertama pada 4x4x4 tidak membutuhkan algoritma untuk dihapal dan lebih bersifat intuitif. Untuk pemula, saya menyarankan untuk membuat blok 1x2 dari dua center piece, dan satu lagi blok 1x2 dari dua center piece sisanya, kemudian menggabungkan kedua blok tersebut menjadi satu center utuh atau blok 2x2. Berikut contoh penyelesaiannya:

	<p>Pada contoh ini dapat kita lihat keempat center piece tersebar di sisi U, F, dan R. Pertama kali, kita akan membuat satu blok 1x2 dengan gerakan: <math>(L' l') U (L l)</math></p> <p><b>PENTING UNTUK DIINGAT:</b> Notasi dengan huruf kecil (<math>f, b, l, r, u, d</math>) berarti memutar satu layer di bagian dalam (inner slice), bukan memutar dua layer sekaligus seperti pada notasi 3x3x3.</p>
	<p>Pada gambar di samping, kita telah membuat satu blok 1x2. Selanjutnya kita akan membuat satu blok lagi dengan gerakan: <math>F2 (D' d')</math>.</p>
	<p>Sekarang saatnya menggabungkan kedua blok tersebut dengan gerakan: <math>(L' l')</math>.</p>
	Center putih telah terbentuk.

Karena bersifat intuitif, kamu bisa bebas mencoba solusi lain. Misalnya seperti contoh berikut ini:

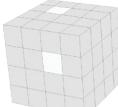
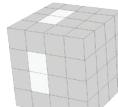
	Ini adalah contoh kasus yang sama seperti contoh sebelumnya. Pertama kali, kita akan membuat satu blok 1x2 dan langsung menggabungkannya dengan piece di layer F dengan gerakan: $(L' L) U' (L' L)$ .
	Pada gambar di samping, kita telah membuat tiga center-piece menjadi satu. Selanjutnya kita akan menggabungkan center piece di sisi R dengan tiga lainnya dengan gerakan: $(D' d') F (D d)$ .
	Center putih telah terbentuk.

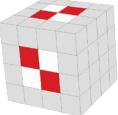
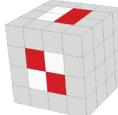
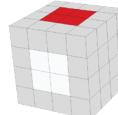
#### ◆ Menyelesaikan Lima Center Berikutnya

Setelah satu center selesai dibentuk, maka kita bisa menentukan warna-warna sisi lainnya sesuai skema warna cube dengan panduan posisi relatif dari center yang telah selesai tersebut. Misalkan pada contoh kita membuat center putih terlebih dahulu, kemudian setelah jadi taruh center putih di bawah (D) dan kita bisa langsung menentukan bahwa kita harus membuat center kuning di sisi atas (U), jika skema warna yang dipakai adalah *western color scheme*.

Kemudian jika setelahnya kita memutuskan untuk membuat center merah di depan (F), maka center orange letaknya di belakang (B), center biru di kiri (L), dan tentulah center hijau di sebelah kanan (R). Kebanyakan cuber mengerjakan center-center yang berlawanan, misalnya dimulai dari center putih kemudian center kuning, merah-oranye, biru-hijau. Kemudian apabila posisi relatifnya terbalik satu sama lainnya, maka letak kedua center tersebut ditukar belakangan. Ada juga cuber yang mengerjakan center yang bersebelahan. Apa pun caranya, kamu yang menentukan gaya permainanmu sendiri. Jadi bebaslah berkreasi. ☺

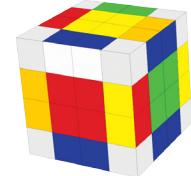
Perlu diingat, bahwa semakin banyak center yang sudah jadi maka semakin terbatas ruang gerak kita. Gerakan yang dilakukan untuk menggabungkan center piece tidak boleh merusak center yang sudah jadi. Di bawah ini saya memberikan contoh-contoh penyelesaian untuk membuat center:

Membuat blok 1x2	
	$(R r) U' (R' r')$
Menggabungkan dua blok 1x2	
	$(R' r') F2 (R r)$

Menyelesaikan dua center		Menukar dua center yang berkebalikan
	$(R\ r) U - (R2\ r2) F\ (R\ r)$  Set gerakan pertama $(R\ r) U$ akan membuat dua blok 1x2 warna putih dan merah. Set gerakan kedua $(R2\ r2) F\ (R\ r)$ akan menghasilkan dua blok putih dan merah lain sekaligus menggabungkan keempatnya.	 $(R2\ r2) F2\ B2\ (R2\ r2)$
	$(R'\ r') F\ (R\ r) - F' - (R'\ r') F2\ (R\ r)$  Set gerakan pertama $((R'\ r') F\ (R\ r))$ akan membuat dua pasang blok 1x2. Kemudian $F'$ untuk meng-align keduanya, lalu set gerakan kedua $(R'\ r') F2\ (R\ r)$ akan menggabungkan keempat blok 1x2 tersebut menjadi center.	
	$(R'\ r') F\ (R\ r)$	
	$(R'\ r') F\ (R\ r)$	
Menukar dua center yang bersebelahan		Menggabungkan Setiap Pasang Double-Edge (Dedge)
	$(R'\ r') F2\ (R2\ r2) U2\ (R'\ r')$	Selanjutnya kita memasangkan ( <i>pairing</i> ) semua dedge sehingga membentuk satu edge seperti pada sebuah 3x3x3. Yang dimaksud dengan dedge atau double-edge adalah sepasang (dua buah) edge yang mempunyai warna sama pada kedua sisinya.

Ada beberapa jenis pairing yang umum digunakan, yaitu satu per satu, dua pair sekaligus, tiga pair sekaligus, dan enam pair sekaligus.

Untuk pemula disarankan menggunakan dan memahami dulu cara pairing satu per satu. Baru setelah itu saya akan menjelaskan metode pairing yang saya pergunakan, yaitu dua pair, tiga pair, dan enam pair sekaligus.



#### ◆ Pairing Satu Per Satu

Dengan cara ini, kita akan melakukan pairing dedge satu edge kemudian menyimpan dedge yang sudah di-pair tadi di layer U atau D, lalu melanjutkan pairing dedge yang lain sampai semuanya selesai. Berikut ini adalah contoh-contoh kasusnya:

No.	Kondisi Awal	Gerakan	Kondisi Akhir
1		(D' d') (L' U' L) (D d)	
2		R2 (D2 d2) (L' U' L) (D2 d2) R2 bertujuan membuat kondisi kedua dedge saling berseberangan dan orientasinya sama.	
3		(R U' R') untuk membuat posisi kedua dedge seperti pada nomor 1, kemudian lakukan gerakan seperti nomor 1: (D' d') (L' U' L) (D d).	
4		U' y (L' U L) y untuk membuat posisi kedua dedge seperti pada nomor 1, kemudian lakukan gerakan seperti nomor 1: (D' d') (L' U' L) (D d).	

#### ◆ Pairing Dua Dedge Sekaligus

Dengan cara ini, kita akan melakukan pairing dedge satu edge kemudian menyimpan dedge yang sudah di-pair tadi di layer U atau D sekaligus menggabungkan satu pair dedge lainnya. Berikut ini adalah contoh-contohnya:

No.	Kondisi Awal	Gerakan	Kondisi Akhir
1		(D' d') y' (R U R') y (D d)	
2		(D' d') (L' U L) (D d)	

#### ◆ Pairing Tiga Dedge Sekaligus

Pertama, kita melakukan putaran tengah cube misalnya (D d) untuk mempersiapkan kondisi pairing. Kemudian men-setup dedge satu demi satu dan pada akhirnya menggabungkan ketiga dedge dengan (D' d') untuk meng-undo gerakan (D d) di awal tadi. Berikut ini penjelasan bagaimana langkah-langkah melakukan pairing tiga dedge (*three-at-a-time*):

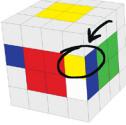
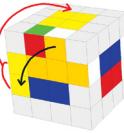
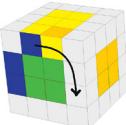
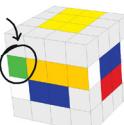
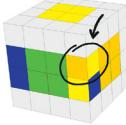
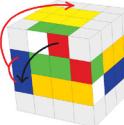
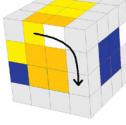
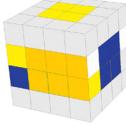
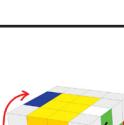
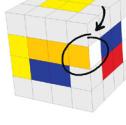
Kondisi yang diperoleh	Gerakan
	<p><b>(D d)</b> Memutar bagian bawah cube sebagai langkah awal.</p>
	<p>Masukkan edge piece putih-biru di Ufl ke FRd dengan gerakan: <b>R U' R'</b>.</p>
	<p>Lakukan rotasi cube ke kanan (<b>y</b>) dan carilah pasangan dari edge piece di sebelah putih-biru tersebut, yaitu warna kuning-biru yang ditunjukkan dengan panah.</p>
	<p>Setelah menemukan pasangannya, masukkan kembali seperti langkah sebelumnya dengan gerakan: <b>R U' R'</b>.</p>
	<p>Lihat kembali piece di sebelahnya, yaitu edge piece warna orange-kuning lalu lakukan rotasi (<b>y</b>) dan carilah pasangan dari piece tersebut.</p>
	<p>Setelah menemukan pasangannya, masukkan kembali seperti langkah sebelumnya dengan gerakan: <b>R U' R'</b>.</p>

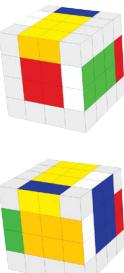
	Langkah terakhir, satukan ketiga dedge dengan gerakan: <b>(D' d')</b>
	Sekarang tiga dedge telah tergabung.

#### ◆ Pairing Enam Dedge Sekaligus

Metode pairing enam sekaligus ini pada dasarnya mirip dengan cara pairing tiga sekaligus. Hanya saja kita langsung men-setup dedge satu demi satu kemudian menggabungkan ketiga dedge dengan **(D' d')** atau **(D d)**, lalu kemudian kembali men-setup tiga pasang dedge berikutnya sambil menyimpan (storing) ketiga pasang dedge yang sudah jadi tadi. Berikut ini penjelasan bagaimana langkah-langkah melakukan pairing enam dedge:

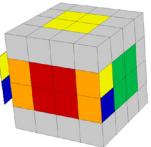
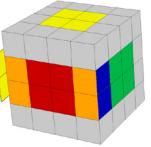
Kondisi yang diperoleh	Gerakan
	<p>Masukkan edge piece putih-biru di Ufl ke FRd dengan gerakan: <b>R U' R'</b>. Perhatikan bahwa kita tidak melakukan gerakan <b>(D d)</b> terlebih dahulu seperti pairing three-at-once.</p>

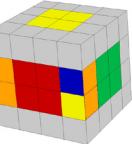
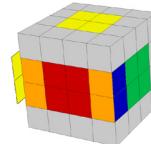
	Lakukan rotasi cube ke kanan ( <b>y</b> ) dan carilah pasangan dari edge piece di sebelah putih-biru tersebut, yaitu warna kuning-biru yang ditunjukkan dengan panah.		Masukkan pasangannya ke posisi yang ditunjuk sehingga dedge kuning-oranye yang sudah jadi "ditendang" keluar ke posisi aman di layer U. Gerakannya: <b>L' U L</b> .
	Setelah menemukan pasangannya, masukkan kembali seperti langkah sebelumnya dengan gerakan: <b>R U' R'</b> .		Lakukan rotasi ( <b>y'</b> ) lalu cari pasangan dari piece yang ditunjuk (hijau-merah).
	Lihat kembali piece di sebelahnya, yaitu edge piece warna oranye-kuning lalu lakukan rotasi ( <b>y</b> ) dan carilah pasangan dari piece tersebut.		Masukkan pasangannya ke posisi yang ditunjuk sehingga dedge biru-kuning yang sudah jadi "ditendang" keluar ke posisi aman di layer U dan digantikan dengan piece tersebut. Gerakannya: <b>L' U L</b> .
	Setelah menemukan pasangannya, masukkan kembali seperti langkah sebelumnya dengan gerakan: <b>R U' R'</b> .		Lakukan rotasi ( <b>y'</b> ) lalu cari pasangan dari piece yang ditunjuk (hijau-putih).
	Satukan ketiga dedge dengan gerakan: <b>(D' d')</b> .		Masukkan piece hijau-putih ke posisi yang ditunjuk dengan gerakan: <b>F U F'</b> . Dedge biru-putih akan berpindah posisi ke layer U.
	Sekarang tiga dedge telah tergabung. Carilah lagi pasangan dari edge piece yang ditunjuk.		

	Pada kondisi ini dapat dilihat bahwa ada tiga pasang dedge yang disimpan di layer atas. Untuk memasangkan tiga dedge sisanya sekaligus mengembalikan center ke posisi semula, lakukan gerakan: <b>(D d)</b> .
	<i>Voila!</i> Enam dedge sekaligus!

#### ◆ Pairing Dua Dedge Terakhir

Untuk dua dedge terakhir, atau jika ditemukan kasus dua pasang dedge yang seperti kasus-kasus di bawah ini:

No.	Kondisi Awal	Gerakan	Kondisi Akhir
1		<b>(D d) R F' U R' F (D' d')</b>	

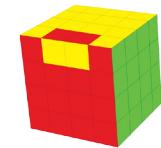
2		Setup sehingga menjadi kondisi 1 dengan gerakan: <b>R F' U F</b> Lalu, lakukan gerakan seperti sebelumnya: <b>(D d) R F' U R' F (D' d')</b> .	
---	---	---	---

### Menyelesaikan 4x4x4 Sebagai Sebuah 3x3x3

Setelah semua center dan dedge selesai dikerjakan, dapat dilihat bahwa bentuk dari 4x4x4 sekarang menyerupai sebuah 3x3x3. Sehingga kita bisa lanjut ke langkah berikutnya yaitu menyelesaikan dengan cara 3x3x3, baik dengan metode beginner atau metode advance (F2L, OLL, dan PLL). Namun terkadang akan dijumpai kondisi "aneh" yang tidak dapat ditemui pada 3x3x3, misalnya satu dedge yang terbalik (flipped) atau dua dedge yang tertukar. Kasus-kasus tersebut dinamakan parity. Ada dua jenis parity, yaitu Orientation Parity dan Permutation Parity.

#### ◆ Orientation Parity

Apabila setelah menyelesaikan semua tahap 3x3x3 pada akhirnya kita menemukan kondisi satu dedge terbalik (flip) seperti gambar di samping, maka berarti kita menghadapi kondisi orientation parity.



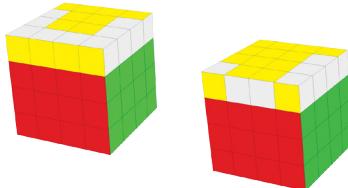
Untuk memperbaikinya, gunakan algoritma ini:

```
r2 B2 U2 l U2 r' U2 r U2 F2 r F2 l' B2 r2
```

**\* PENTING UNTUK DIINGAT:**

Notasi dengan huruf kecil (f, b, l, r, u, d) berarti memutar satu layer di bagian dalam (inner slice), bukan memutar dua layer sekaligus seperti pada notasi 3x3x3.

Sebenarnya pada saat kita selesai membuat F2L pada 4x4x4 sudah bisa dideteksi apakah ada kasus orientation parity, yaitu dengan menghitung jumlah edge sisi atas yang sudah ter-orientasi. Apabila jumlahnya ganjil (1 atau 3) berarti ada orientation parity, seperti gambar di bawah ini:



Perhatikan dedge kuning yang sudah ter-orientasi pada gambar di atas. Pada gambar pertama, hanya ada satu dedge yang ter-orientasi, sedangkan pada gambar kedua ada tiga yang sudah ter-orientasikan. Pada kedua kasus tersebut terdapat orientation parity. Pada tahap ini kita sudah bisa melakukan algoritma parity fixing dengan lebih mudah. Yaitu dengan memutar dua layer sekaligus (wide layer

turn) dan bukannya memutar inner slice. Tentunya eksekusi algoritma akan menjadi lebih mudah dan cepat, tetapi dengan hasil akhir yang sama. Baru setelah itu kita melakukan OLL seperti biasa.

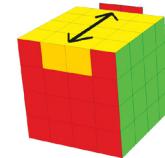
Algoritma untuk memperbaiki orientation parity yang dimodifikasi menggunakan wide turn:

```
(R2 r2) B2 U2 (l l) U2 (r' r') U2 (R r)  
U2 F2 (R r) F2 (l' l') B2 (R2 r2)
```

◆ **Permutation Parity**

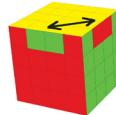
Kadang-kadang juga kita akan menemukan kasus dua dedge yang tertukar (50% kemungkinan). Kasus ini dinamakan permutation parity.

Pada gambar di samping, dua dedge yang saling berseberangan tertukar satu sama lain. Untuk membetulkannya, gunakan salah satu dari algoritma berikut:



```
r2 U2 r2 (U2 u2) r2 u2  
atau  
(U2 u2) (R2 r2) U2 r2 U2 (R2 r2) (U2 u2)
```

Apabila kasus yang kamu temui adalah seperti gambar di sebelah, yaitu dua dedge yang bersebelahan tertukar satu sama lain, maka lakukan setup move. Misalkan F U' F, untuk membawa dedge di atas-kanan ke posisi atas-belakang. Setelah itu selesaikan dengan algoritma yang sama:



**r2 U2 r2 (U2 u2) r2 u2**  
atau  
**(U2 u2) (R2 r2) U2 r2 U2 (R2 r2) (U2 u2)**

*Metode Solving*  
**5x5x5**

## 1. Pengantar

Cube 5x5x5 merupakan cube “ganjil” dikarenakan dimensi panjang, lebar, dan tingginya adalah 5 yang merupakan angka ganjil. Pada 5x5x5 terdapat center yang *fixed* atau tetap pada posisinya, sama seperti pada 3x3x3. Sehingga untuk mengetahui warna dari suatu sisi, caranya juga sama seperti 3x3x3 yaitu dengan melihat warna pada center piece.

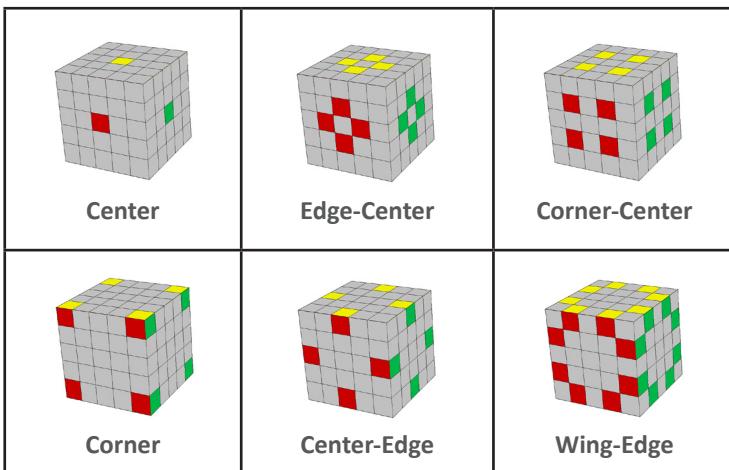
### Tahukah kamu?

Jumlah kemungkinan posisi dari sebuah cube 5x5x5 amat sangat mencengangkan, yaitu 280 trevigintillion ( $1,8 \times 10^{72}$ )!

Saat ini, rekor dunia 5x5x5 (single solve) dipegang oleh Dan Cohen dari Amerika Serikat, dengan catatan waktu 1 menit 7,25 detik. Dan Cohen juga memegang rekor dunia untuk average of 5 dengan waktu 1 menit 13,36 detik.



Cube 5x5x5 memiliki beberapa jenis piece, seperti dijelaskan oleh gambar di bawah ini:

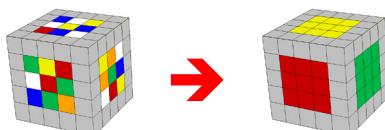


## 2. Metode Menyelesaikan 5x5x5

Metode solving 5x5x5 pada dasarnya sama dengan cube 4x4x4 yaitu membuat bentuk atau susunan 5x5x5 menjadi seperti 3x3x3 kemudian menyelesaiakannya dengan metode solving 3x3x3 (Reduction method).

Langkah-langkahnya adalah sebagai berikut:

### Menyelesaikan Center

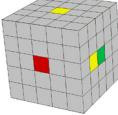
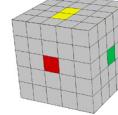
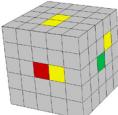
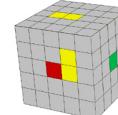


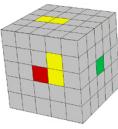
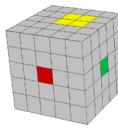
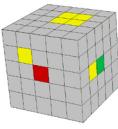
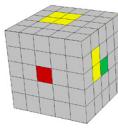
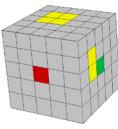
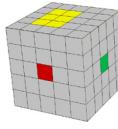
Langkah pertama adalah menyelesaikan semua center piece/cubies yang berjumlah 54 buah sehingga menyerupai center pada sebuah 3x3x3.

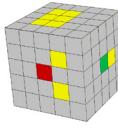
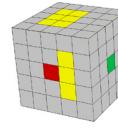
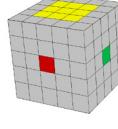
#### ◆ Menyelesaikan Center Pertama

Tugas pertama dalam membuat center adalah membuat blok 1x2, dengan cara mempertemukan sebuah edge-center dengan center yang sewarna.

Berikut contoh penyelesaiannya:

No.	Kondisi Awal	Gerakan	Kondisi Akhir
1		(F' f')	
2		(U u) Membuat blok 1x2 dari edge-center di R dengan corner-center di F.	

3		<b>U (R r)</b> Meng-align blok tersebut dengan blok 1x2 di U, kemudian (R r) untuk menggabungkan keduanya menjadi blok 2x2.	
4		<b>(U' u')</b> untuk membuat blok 1x2 dari corner-center di F dan edge-center di R.	
5		Kemudian lakukan <b>(F' f')</b> untuk menggabungkan blok 1x2 di R dengan blok 2x2 di U sehingga menjadi satu blok 2x3	

6		<b>E'</b> untuk membuat blok 1x3 (garis), kemudian <b>(R r)</b> untuk menggabungkannya dengan blok 2x3 di U.	
7		Satu center telah selesai.	

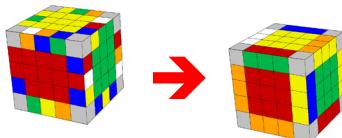
#### ◆ Menyelesaikan Lima Center Berikutnya

Perlu diingat, bahwa semakin banyak center yang sudah jadi maka semakin terbatas ruang gerak kita. Gerakan yang dilakukan untuk menggabungkan center piece tidak boleh merusak center yang sudah jadi. Disarankan untuk membuat dua center yang saling berkebalikan terlebih dahulu, kemudian untuk empat center berikutnya kerjakanlah center-center yang letaknya bersebelahan. Sisakanlah dua center terakhir untuk disolve yang letaknya bersebelahan. Menyelesaikan dua center terakhir dengan cara ini akan mempermudah kita melakukan gerakan dan pengenalan piece-piece-nya. Misalnya rute kita untuk solving center adalah: mengerjakan center kuning terlebih dahulu kemudian center putih, selanjutnya center merah dan hijau, lalu terakhir kita selesaikan center biru dan oranye.

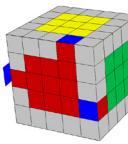
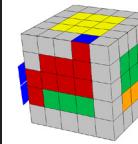
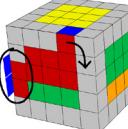
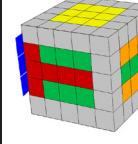
Pola-pola yang sering ditemui dalam menyelesaikan dua center terakhir:

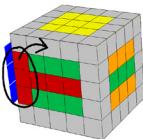
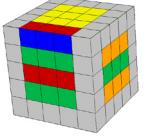
No.	Kondisi	Gerakan
1		$(R r) U M' U' (R' r') U M$
2		$(R r) U (R' r') U (R r) U2 (R' r')$ Gerakan ini mirip dengan algoritma "Sune" pada 3x3x3.

### Menggabungkan Triple-Edge (Tredge)

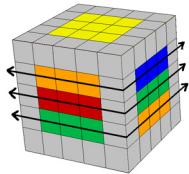


Pada tahap ini, kita akan memasangkan semua wing-edge dengan center-edge sehingga membentuk satu edge seperti pada sebuah 3x3x3 (tredge). Peganglah cube dengan sisi kuning di atas sampai kita selesai membuat 4 tredge. Cara yang kita gunakan di sini disebut dengan "storage system", yaitu membuat tredge di layer tengah (sebenarnya tiga layer tengah!) lalu menyimpannya di layer U dan D.

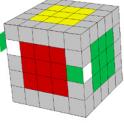
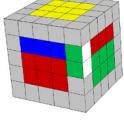
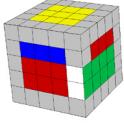
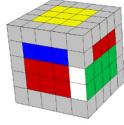
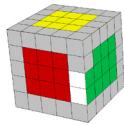
No.	Kondisi Awal	Gerakan	Kondisi Akhir
1		$(D' d')$  Pertama, kita memasangkan wing-edge di FRd dengan center edge di FL. * Biarkan posisi center seperti pada kondisi akhir di kanan, nanti kita akan membetulkannya setelah selesai membuat tredge ke 8.	
2		Masukkan wing-edge di sticker FUr ke posisi Fru tanpa mengubah posisi center tadi dengan gerakan yang mirip dengan 4x4x4, yaitu: $R U R'$ . Kemudian gabungkan dengan dua edge lainnya dengan gerakan $U$	

		Masukkan wing-edge di FUr ke posisi FrU tanpa mengubah posisi center tadi dengan gerakan yang mirip dengan 4x4x4, yaitu: <b>R U R'.</b> Kemudian gabungkan dengan dua edge lainnya dengan gerakan <b>U</b> .	
3			

Lanjutkan memasangkan setiap tredge. Setelah selesai memasangkan tredge keempat dan layer U sudah penuh dengan tredge yang disimpan di sana, putar cube sehingga layer U menjadi D dan D menjadi U. Buat empat tredge berikutnya dan simpan di layer U. Tidak perlu mengembalikan center yang teracak, selama kita menjaganya dalam kondisi teracak horizontal seperti pada gambar di samping. Setelah layer U dan D terisi penuh dan tersisa 4 tredge yang belum diselesaikan barulah kita membetulkan center.

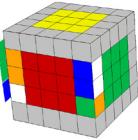
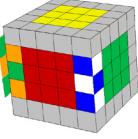


Untuk empat tredge terakhir, ada berikut ini contoh gerakan dasar untuk memasangkan center-edge dengan wing-edge:

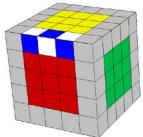
No.	Kondisi Awal	Gerakan	Kondisi Akhir
1		(U' u')	Untuk menggabungkan center-edge putih-hijau dengan wing-edge putih-hijau.
2		Kemudian lakukan gerakan berikut untuk menge-flip kedua edge piece tersebut dan meletakannya di bawah slice u dengan gerakan: (R U R') U' (F' U F).	
3		Kembalikan center ke posisinya semula dengan gerakan (U u).	

#### ◆ Dua Tredge Terakhir

Berikut ini adalah beberapa kondisi yang dapat ditemui saat solving dua tredge terakhir:

No.	Kondisi Awal	Solusi
1		$(U' u') F2 (U' u') F2 - L2 (U' u')$ $L2 (D d) F2 (D' d') F2 (U2 u2)$
2		$E (R U R') U' (F' U F) E'$

Ada sekitar 50% kemungkinan kamu akan menghadapi kondisi tredge terakhir ter-flip seperti gambar di samping ini. Untuk menyelesaiakannya, kita dapat menggunakan algoritma untuk membetulkan orientation parity pada 4x4x4, yaitu:

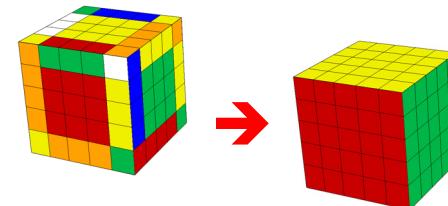


**r2 B2 U2 | U2 r' U2 r U2 F2 r F2 l' B2 r2**

Atau:

**(R2 r2) B2 U2 (L l) U2 (R' r') U2 (R r)**  
**U2 F2 (R r) F2 (L' l') B2 (R2 r2)**

### 3. Menyelesaikan 5x5x5 Sebagai Sebuah 3x3x3



Setelah semua center dan tredge selesai dikerjakan, sekarang kita bisa menyelesaikan 5x5x5 seperti layaknya menyelesaikan sebuah 3x3x3. Pada tahap ini tidak akan ditemui lagi kondisi parity seperti pada 4x4x4.

*Metode Solving*  
 $2 \times 2 \times 2$

## 1. Pengantar

Bagi kebanyakan orang awam, cube 2x2x2 nampak mudah untuk diselesaikan. Namun begitu, *puzzle* ini juga memiliki kombinasi posisi yang cukup banyak meski tak sebanyak 3x3x3 yaitu (hanya) 3.674.160 kemungkinan. Sering saya menemui orang awam yang menganggap remeh si Kecil satu ini. Tetapi begitu mencobanya beberapa saat, mereka menyerah. Saya sering menggunakan 2x2x2 sebagai alat “pemanasan” ketika melakukan demo speedsolving. Kebanyakan orang awam dengan gagah berani menerima tantangan untuk menyelesaikan 2x2x2 begitu saja, namun berakhir dengan kegagalan. ☺

## 2. Menyelesaikan 2x2x2 dalam Tiga Langkah Mudah

Bagi cuber yang telah sanggup menyelesaikan 3x3x3, tentunya akan bisa menyelesaikan 2x2x2 dengan mudah. Namun bagi pemula yang belum bisa solving 3x3x3 pun dapat mengikuti panduan langkah-langkah solving 2x2x2 ini.



Pada prinsipnya sebuah 2x2x2 adalah sama seperti 3x3x3, hanya minus center piece dan edge piece dan menyisakan delapan buah corner saja.

Oleh karena itu, metode solving yang saya terangkan di sini adalah cara layer-by-layer (LBL) yang mengadopsi metode solving 3x3x3. Metode ini cocok untuk pemula.

### Menyelesaikan Layer Pertama

Tujuan dari langkah pertama ini adalah membuat satu lapisan seperti gambar di samping. Berbeda dengan 3x3x3, pada 2x2x2 yang tidak memiliki center dan edge piece kita langsung membentuk satu layer dengan menempatkan keempat corernya pada posisi yang benar. Caranya mirip dengan langkah kedua pada bab 3x3x3 pemula. Pada langkah ini, kita menyelesaikan layer pertama di sisi bawah (D), dengan maksud untuk meminimalisasi cube rotation dan penggunaan fingertrick lebih mudah karena kita memutar sisi atas dengan jari telunjuk.



Berikut ini adalah kondisi-kondisi yang ditemui pada tahap ini beserta cara menyelesaikannya:

#### Kasus 1: Corner putih ada di layer atas

	F' U' F
	R U R'
	(R U' R') U2 (R U R')

#### Kasus 2: Corner putih ada di layer bawah, tetapi orientasinya salah

	RUR' U' untuk membawa corner ke layer atas, kemudian selesaikan seperti pada kasus 2.
--	---

### Menyamakan Warna Sisi Atas

Langkah ini diadopsi dari tahap kelima dari metode pemula 3x3x3. Pada langkah ini, kita melihat jumlah stiker sisi atas yang sudah dan belum ter-orientasi. Perhatikan arah stiker

kuning di lapis atas, lalu posisikan cube sesuai pola-pola di bawah, kemudian lakukan algoritma:

**(R U) (R' U) (R U2 R')**

Apabila belum semua stiker ter-orientasi atau sisi atas belum kuning semua, cocokkan posisinya kembali dengan pola di bawah lalu lakukan lagi algoritma tadi. Mungkin kamu harus beberapa kali mengeksekusi algoritma tersebut sebelum sisi kuning terbentuk sempurna.

#### Kasus 1: Tidak ada stiker kuning yang menghadap atas



Pegang cube dengan satu stiker kuning di corner kiri-atas-depan (LFU) menghadap ke kiri.

#### Kasus 2: Satu stiker kuning menghadap atas



Letakkan posisi stiker kuning yang sudah ter-orientasi di kiri-atas-depan

#### Kasus 3: Dua stiker kuning yang sudah jadi



Apabila ada dua corner yang sudah ter-orientasi, pegang cube dengan stiker kuning di atas-kiri-depan menghadap ke depan.

#### Menempatkan Corner ke Posisi yang Benar (Permute Corners)

Pada tahap terakhir solving 2x2x2 ini, kita akan menemukan kondisi di mana ada dua corner yang tertukar, yaitu corner yang bersebelahan dan corner yang letaknya berseberangan secara diagonal. Untuk menyelesaiakannya, kita menggunakan algoritma permutasi yang diadopsi dari algoritma PLL pada cube 3x3x3.

Apabila corner yang ingin ditukar letaknya bersebelahan, seperti pada gambar di samping, maka lakukan algoritma:



**(R U2) (R' U') (R U2) (L' U) (R U' L)**

Apabila kedua corner yang ingin ditukar letaknya berseberangan secara diagonal, maka lakukan algoritma di atas dua kali, kalau kamu belum mau menghapal algoritma baru. Atau lakukan algoritma berikut ini sekali saja:



**(F R U') (R' U' R U) (R' F')  
(R U R' U') (R' F R F')**

### 3. Metode Tingkat Lanjut (Ortega Method)

Metode ini terdiri dari tiga langkah yaitu menyamakan warna satu sisi, lalu menyamakan warna sisi sebaliknya, kemudian membetulkan semuanya dalam satu kali jalan. Metode Ortega relatif lebih cepat daripada Layer-By-Layer, karena cenderung lebih mudah untuk membuat warna satu sisi daripada satu lapis.

#### Menyamakan Warna Satu Sisi

Membuat warna pada satu sisi, misalnya sisi D menjadi sama. Ingat, **BUKAN** membuat **SATU LAYER** melainkan **HANYA SATU SISI** saja.



#### Menyamakan Warna Sisi Sebaliknya

Tahap ini identik dengan tahap mengorientasikan corner pada 3x3x3 (2-Look OLL). Kamu bisa menggunakan algoritma 2-Look OLL tersebut atau menggunakan algoritma yang ada di sini (beberapa memiliki gerakan yang lebih sedikit). Ada tujuh kasus, yaitu:

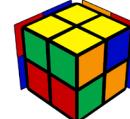
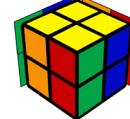
$(R' U' R U')(R' U2 R)$	$(R' U2)(RUR'UR)$	$(FR'F')(RUR'U'R')$	$(RUR'U'')(R'FRF')$

$F(RUR'U')F'$	$F(RUR'U')(RUR'U')F'$	$R2 U2 R U2 R2$	

#### Menyelesaikan Semuanya

Pada tahap terakhir ini, kita akan menyelesaikan seluruh cube dalam satu langkah saja.

			
$\text{RU2R}'\text{U}' \text{RU2}$ $\text{L}'\text{UR}'\text{U}'\text{L}'$	$\text{RU}'\text{R}'\text{U}' \text{F2}$ $\text{U}'\text{UR}' \text{DR2}$	$\text{R2U F2 U2}$ $\text{R2UR2}'$ $y2 \text{R2 U R2 } y'$ $\text{U2 R2 U R2 }$	$\text{R2 U R2 } y'$ $\text{U2 R2 U R2 }$
			
$\text{R2 U R2 } y'$ $\text{U2 R2 U R2 }$ $\text{U2}$	$\text{L D}' \text{L F2 L}'$ $\text{D L}'$	$\text{R U}' \text{R F2 R}'$ $\text{U R}'$	$\text{R2 F2 R2}$

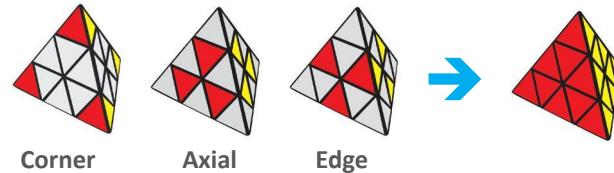
Cara mengenali kasus-kasus di atas sebaiknya dengan melihat blok warna 1x2 yang kita sebut "bar". Hanya ada satu kasus yang tidak memiliki bar, yaitu kasus ke delapan. Sedangkan pada kasus ketiga ada dua bar sewarna di layer atas dan layer bawah.



## Metode Solving Pyraminx

### 1. Mengenal Pyraminx dan Strukturnya

Pyraminx merupakan *twisty puzzle* berbentuk *tetrahedron* yang memiliki 4 sisi. Bentuknya mirip dengan piramid. Pyraminx memiliki 4 corner tip, 4 axial piece, dan 6 edge piece. Corner merupakan sudut-sudut dari Pyraminx yang memiliki 3 sisi, sedangkan Axial adalah layer di bawah corner yang memiliki 3 sisi. Corner dan axial sebetulnya merupakan satu kesatuan dimana keduanya berfungsi sebagai poros dari Pyraminx. Edge adalah bagian rusuk dari Pyraminx yang memiliki 2 sisi. Bila sebuah Pyraminx diacak, maka edge piece dapat berpindah-pindah posisi namun axial dan corner hanya berputar pada tempatnya saja (mirip center piece pada 3x3).



## 2. Menyelesaikan Pyraminx Layer-By-Layer

### Membetulkan Corner Layer Pertama (Base Layer)

Langkah pertama ini bertujuan untuk membetulkan orientasi dari corner beserta axial piece pada layer pertama (base layer), seperti gambar di samping.



Putarlah corner sehingga warna ketiga sisi pada corner bertemu dengan warna-warna pada axial piece, lalu putarlah keduanya menuju satu sisi, sehingga membentuk logo "Mitsubishi" seperti pada gambar. Nah, untuk langkah pertama ini, coba kamu buat logo "Mitsubishi" warna biru.

Cara untuk menentukan warna suatu sisi yaitu dengan melihat warna-warna yang ada pada suatu corner. Bila kamu ingin mencari di mana sisi biru seharusnya, carilah corner yang **tidak memiliki stiker berwarna biru** maka sisi biru pasti ada di seberang corner tersebut. Putarlah ketiga corner beserta axial piecennya sehingga bertemu pada satu sisi dan membentuk logo "Mitsubishi". Atau bisa juga dengan cara **melihat dari ketiga corner yang memiliki warna biru** pada satu layer, maka berarti sisi biru berada di layer tersebut.

Langkah ini mirip dengan membuat cross pada Rubik's Cube 3x3, serta sangat mudah dan dapat dilakukan dengan intuitif (tanpa algoritma).

### Menyelesaikan Pondasi Piramid



Setelah selesai membentuk "Mitsubishi" pada sisi biru, maka letakkan sisi tersebut menghadap ke bawah (D). Kemudian identifikasi edge-edge yang seharusnya berada di layer bawah (yang memiliki stiker berwarna biru) dan masukkan ke slotnya yang benar. Tujuan dari langkah ini adalah membentuk layer paling bawah dari piramida kita. Langkah ini mirip dengan langkah kedua dari metode beginner 3x3 dan dapat dilakukan juga secara intuitif.

#### Kasus 1: Edge biru ada di layer kedua

	$R U' R'$
	$L' U L$
	$L R' L' R$
	$R' L R L'$

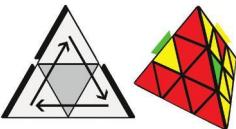
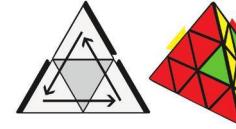
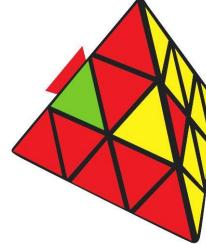
	L' U' L
	R U R'
<b>Kasus 2: Edge biru ada di layer bawah, tetapi bukan pada tempat yang benar</b>	
	Bawa piece ke layer kedua dengan gerakan <b>R U R'</b> atau <b>R U' R'</b> , lalu selesaikan dengan gerakan sesuai dengan salah satu kondisi dari kasus 1.

Kode	Pola	Algoritma Solusi	Cara Membuat
X1		<b>R U R' U</b> <b>R U R'</b> atau <b>Rw' R L R'</b> <b>L R L R'</b>  * Algoritma ini mirip dengan "Sune" pada 3x3	<b>R U' R' U'</b> <b>R U' R'</b>
X2		<b>R' U' R U'</b> <b>R' U' R</b> atau <b>Rw' R L' R'</b> <b>L' R L' R'</b>  * Inverse dari X1	<b>R' U R U</b> <b>R' U R</b>

## Menyelesaikan Puncak Piramid

Setelah sisi biru selesai, kemudian kamu betulkan dahulu posisi corner dan axial piece pada layer atas Pyraminx sehingga tampak seperti gambar di samping. Setelah itu akan tersisa tiga buah edge yang belum ter-solved yang dapat diselesaikan dengan 5 algoritma (2 di antaranya mirror dan inverse dari algoritma yang lainnya).



Y1	<p><math>R' L R L'</math>  <math>U' L' U L</math>  atau  <b><math>R' L R L' L w</math></b>  <b><math>R' L' R L</math></b></p> <p>Cara pe-nge-nala-nan pola ini bisa dilihat dari adanya dua stiker yang letaknya saling ber-lawanan pada satu layer (stiker kuning) dan adanya dua stiker se-warna (merah).</p> <p><math>L' U' L U</math>  <math>L R' L' R</math></p> 		<p>Y2</p>  <p><math>L R' L' R</math>  <math>U R U' R'</math>  atau  <b><math>L R' L'</math></b>  <b><math>R R w' L</math></b>  <b><math>R L' R'</math></b></p> <p>* Mirror dari X1</p>	<p><math>R U R' U</math>  <math>R' L R L'</math></p>
		<p>Z</p> 	<p><math>L R' L' R</math>  <math>U' R U R'</math>  atau  <b><math>L R' L'</math></b>  <b><math>R R w' L'</math></b>  <b><math>R L R'</math></b></p> <p>* Disebut juga algoritma "Flipper", yang akan membalik 2 edge.</p>	<p><math>R U' R' U</math>  <math>R' L R L'</math></p>

Sekarang pasti Pyraminx kamu sudah solved!



## Tip Mempercepat Solving Pyraminx

### ◆ Extended “Mitsubishi”

Pada saat kita melakukan inspeksi, kita sekaligus mencari target edge pertama dan memasukkannya sekaligus pada saat membuat Mitsubishi. Prinsipnya mirip dengan Extended Cross pada 3x3.

Untuk lebih jelasnya, coba scramble Pyraminx kamu dengan algoritma di bawah ini dengan posisi warna merah di depan (F) dan biru di bawah (D):

**r b u' - B L' B' - U L' - R U' R' U - B U B' - L R' - B U R B' - R'  
L R L'**

Kemudian saya membetulkan tiga corner pada base layer dengan: **r' b' U' L**

Perhatikan notasi yang dicetak tebal. Saya menambahkan **U'** ketimbang hanya melakukan **r' b' L** untuk membetulkan corner dan axial piece. Dengan **U'** saya bisa memasukkan satu edge piece berwarna biru-merah berbarengan dengan proses membetulkan corner terakhir. Dengan demikian, kita hanya perlu memasukkan 2 edge sisanya ke base layer. Lumayan menghemat jumlah gerakan bukan?

### ◆ Looking Ahead dan Multiple Angle

Sama seperti saat melakukan F2L pada Rubik's Cube, pada Pyraminx juga bisa diterapkan hal yang sama, yaitu pada saat memasukkan satu edge ke posisinya kita mencari edge berikutnya. Ketika slotting edge ke posisinya kita juga tidak perlu sering-sering merotasi Pyraminx. Contoh:

Lanjutkan scramble dari bagian (A) di atas. Setelah kerangka bawah terbentuk, masukkan edge kedua ke kiri bawah dengan gerakan **y' L' U L**. Tanpa mengubah posisi, lanjutkan dengan memasukkan edge berikutnya ke kanan bawah dengan **L U' L'**. Atau lakukan slotting kedua edge tersebut dengan satu gerakan tidak terputus: **y' L' U L' U' L'**. Dapat dilihat bahwa dengan cara kedua yang lebih efisien, kita tidak membuang-buang gerakan dengan melakukan **y' L' U LL U' L** (perhatikan yang digaris bawah). Daripada memutar **L** dua kali, kita memutar **L'** yang pada dasarnya sama saja.

## *Metode Solving Blindfold*

### **Blindfold Cubing**

Blindfold cubing atau sering disingkat dengan BLD, adalah suatu cabang dari permainan Rubik, yaitu menyelesaikan Rubik dengan mata tertutup (blindfolded). Ada beberapa metode BLD yang populer digunakan oleh para BLD-ers, seperti Old Pochmann, 3OP (3 Cycle Orient and Permute), M2/R2, BH (Beyer-Hardwick), dan metode yang saya kembangkan, yaitu ABC (Abel Brata's Cycle).

Sebelum memulai penjelasan di bab ini, saya tekankan sekali lagi, metode BLD yang saya jelaskan di sini **bukanlah untuk cuber pemula**. Seperti kata pepatah, jangan coba berlari sebelum kamu bisa berjalan! Kalau kamu masih baru mulai mengenal cubing, saya sarankan pelajarilah dahulu bab-bab lain mengenai speedcubing sebelum kamu me-langkah memasuki bab ini.

## Metode ABC (Abel Brata's Cycle)

Metode ini adalah metode blindfolded cubing yang saya kembangkan sendiri. Dengan metode ini, kita menyelesaikan 2 edge atau 3 edge sekaligus dalam sekali jalan. Dasar dari metode ini adalah men-cycle stiker, bukannya piece seperti pada metode 3OP (3-Cycle Orient and Permute). Tujuan dari penciptaan metode ini adalah efektivitas jumlah gerakan dan mengurangi beban memorisasi, karena kita hanya perlu menghapal rute cycle stiker untuk edges dan corners, tanpa perlu memecah memorisasi menjadi 4 tahap seperti 3OP (Corner Orientation, Corner Permutation, Edge Orientation, Edge Permutation).

Sebenarnya **metode ini merupakan pendekatan saya terhadap metode cycle dengan cara lain**. Ada yang menganggap metode ini termasuk freestyle, karena seringkali saya tidak menggunakan fixed buffer atau piece/stiker yang dijadikan tempat untuk menampung item yang akan diselesaikan berikutnya (kecuali kalau break-a-new-cycle, akan ada fixed buffer, yaitu item pertama).

Oke, mari kita mulai.

Yang perlu kamu ketahui untuk mempelajari metode ini adalah Permutasi U (untuk solving edges) dan Permutasi A (untuk corners) dan saya yakin kamu pasti sudah hafal luar kepala algoritmanya. Selain itu, untuk membetulkan kasus-kasus parity kamu bisa menggunakan salah satu dari 21 algoritma PLL (tidak perlu menghapal semua PLL, cukup menggunakan yang kamu tahu saja).

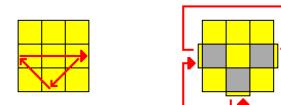
Apabila kamu sudah mahir melakukan setup permutasi U dan A secara intuitif, kamu bisa melanjutkan dengan mempelajari algoritma-algoritma TuRBO method ataupun menggunakan algoritma 5-cycle.

Pada suatu kondisi stiker teracak, biasanya terdapat beberapa cycle. Dengan begitu, kita harus mengingat setiap cycle secara terpecah, atau dengan kata lain tidak ada fixed buffer seperti metode Pochmann atau M2. Tetapi jika dirasa dengan cara demikian beban memorisasi menjadi berat, kamu bisa menggunakan metode break-into-new-cycle. Sehingga rute stiker yang harus diingat menjadi satu rute panjang, bukan rute cycle yang terpecah-pecah. Penjelasan mengenai ini akan kita bahas nanti.

### ◆ Menyelesaikan Edge

Untuk menyelesaikan edge, kita menggunakan permutasi U. Sebelum men-cycle edge, kita harus men-setup stiker-stiker dari edge yang ingin di-cycle dengan dua rule: (1) semua stiker diletakkan di satu permukaan sisi yang sama, (2) semua stiker diletakkan di samping sisi pada layer dimana edge akan di-cycle.

Permutasi U Clock Wise (CW):



Permutasi U Counter Clock Wise (CCW):



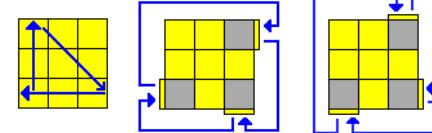
Contoh kasus dan gerakan setup-nya:

No.	Cycle	Setup	Kondisi Akhir
1		$d R'$	
2		$F' R F$	
3		$D' L2$	

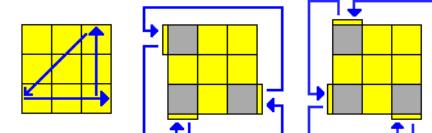
#### ◆ Menyelesaikan Corner

Sama seperti pada edge, untuk corner kita harus melakukan setup stiker-stiker yang ingin di-cycle seperti diagram di bawah ini:

Permutasi A Clock Wise (CW):



Permutasi A Counter Clock Wise (CCW):



Contoh kasus dan gerakan setupnya:

No.	Cycle	Setup	Kondisi Akhir
1		$R$	
2		$B'$	

### ◆ Parity

Terkadang apabila ditemui suatu cycle yang berjumlah genap (misalnya cycle edge yang panjangnya 4, 6, 8, atau 10 stiker), maka akan ditemui sisa edge yang membentuk 2-cycle (saling bertukar). Sehingga untuk menyelesaikannya kita harus memasangkan dengan 2-cycle lainnya dan men-setup stikernya sesuai kasus PLL.

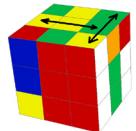
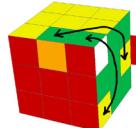
Contohnya: 2 edge dan 2 corner, dapat diselesaikan dengan algoritma PLL yang menukar 2 edge dan 2 corner seperti T, R, J, Y, V. Atau jika tersisa dua pasang 2-cycle edge, maka PLL yang bisa digunakan adalah H dan Z.

Pada contoh kasus di samping, edge hijau-orange di UF tertukar dengan kuning-merah di RB dan corner hijau-merah-putih di UFR tertukar dengan corner hijau-merah-kuning di DFR. Untuk menyelesaikannya, saya setup sehingga keempat piece tersebut membentuk permutasi T dengan gerakan: **B' U' F'**.

Maka posisi piece akan menjadi seperti gambar di samping. Yang dapat diselesaikan dengan permutasi T:

**(R U R' U') (R' F) (R2 U') (R' U' R U) (R' F')**

Setelah itu, jangan lupa undo setup dengan gerakan: **F' U B'**.



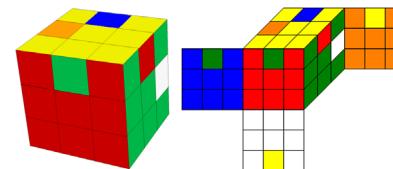
### ◆ Teknik Tambahan

#### Breaking into a new cycle

Misalkan kita mempunyai dua cycle (AB)(CDEF). Dimana (AB) dan (CDEF) berjumlah genap (akan didapat dua 2-cycle pada akhir eksekusi). Mengeksekusi (ABC) akan memecah cycle kedua dan kita akan mendapat cycle tersebut menjadi (ADEFC), dimana C diletakkan di akhir cycle. Teknik ini akan membantu kita untuk menghindari adanya 2-cycle dalam cycle berjumlah genap. Bisa juga digunakan untuk menyelesaikan double-transposition, misalnya (AB)(CD) yang bisa kita selesaikan dengan 2 kali 3-cycle yaitu menjadi (ABC)(ADC).

#### Contoh:

Pegang cube kamu dengan warna merah di depan dan kuning di atas (bila menggunakan standard color scheme). Acak cube kamu dengan gerakan **R B2 R' B' L' U' B' U L B L U' L'** untuk mendapatkan kondisi cube seperti di bawah ini:



Pada contoh ini, kita mendapatkan kasus cycle (UF-UR) (UB-LU-RB-DB). Yang pertama adalah 2-cycle atau dua edge piece tertukar. **Kita break ke cycle berikutnya**, sehingga cycle menjadi (UF-UR-UB) kemudian (UF-LU-RB-DB-UB).

Pertama kali kita akan menyelesaikan cycle (UF-UR-UB):

### **U – Permutasi U CCW – U'**

Setelah itu kita lanjut untuk menyelesaikan cycle (UF-LU-RB-DB-UB):

- Cycle 1 (UF-LU-RB): **L R2 x U2 – Permutasi U CCW – U' x' R2 L'**
- Cycle 2 (UF-DB-UB): **D' R2 y – Permutasi U CCW – y' R2 D**

### **Pick-Up Cycling**

Mungkin adakalanya kita lupa satu stiker dari sebuah cycle. Misalnya kita punya cycle (ABCDEFG) dan lupa di mana letak B. Yang kita ingat adalah (ACDEFG), maka kita melakukan (ACD) dan (AEF) untuk mereduksi cycle menjadi (AG). Melewatkannya B yang sebenarnya membuat suatu cycle tambahan yang disebut “pick-up cycle” (BC), yang terbentuk dari stiker yang kita lupakan dan stiker dengan urutan setelahnya dari cycle semula (ABCDEFG).

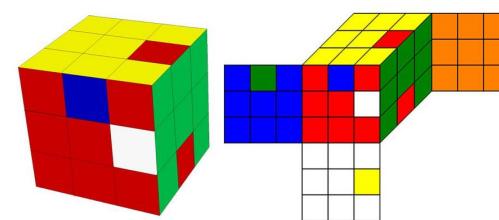
Kemudian kita dapat menyelesaikan (AG)(BC) dengan double-transposition.

Teknik ini juga berlaku jika kita lupa lebih dari satu stiker. Contohnya, kita punya cycle (ABCDEFGHIJ) dan kita lupa dimana B, E, F, dan I. Cycle yang kita ingat menjadi (ACDHGJ), kemudian kita selesaikan (ACD) dan (AGH) sehingga tersisa (AJ). Selanjutnya, kita harus menyelesaikan tiga pick-up cycle tambahan yaitu (BC), (EFG), dan (IJ). Dimana (IJ) harus diselesaikan setelah (AJ), yang sama saja dengan melakukan (AIJ).

Teknik ini digunakan saat kita tidak dapat mengingat satu stiker dalam suatu cycle. Daripada berhenti mengeksekusi dan berusaha keras mengingat-ingat satu stiker tersebut, kita selesaikan saja seluruh cycle dan kembali ke pick-up cycle ketika kita telah ingat letak stikernya. Satu trik lagi... yaitu memasangkan atau menggenapkan cycle (pairing off). Misalkan kita lupa satu stiker dalam cycle berjumlah ganjil (odd cycle), contohnya kita lupa B dari cycle (ABCDE). Mengeksekusi (ACD) membuat adanya double-transposition (AE)(BC), yang lebih sulit diselesaikan daripada 3-cycle. Sebaliknya dengan menyelesaikan (ADE) dahulu, kita akan memperoleh cycle (BCD). Jadi daripada melanjutkan langsung ke (ACD), kita memasangkan B dengan satu stiker sesudahnya. Cara kedua ini lebih mudah dilakukan dari cara pertama yang menyisakan double-transposition.

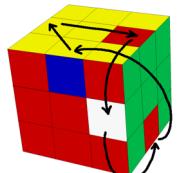
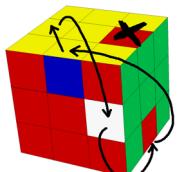
### **Contoh:**

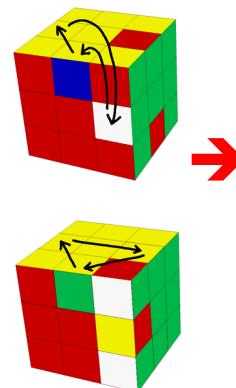
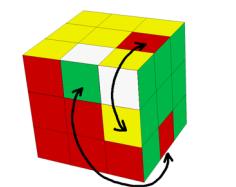
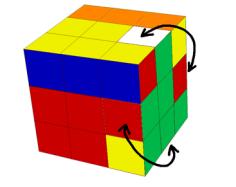
Pegang cube kamu dengan warna merah di depan dan kuning di atas (bila menggunakan standard color scheme). Lakukan gerakan **D2 B2 R' F2 U' R2 B2 L2 D F2 L F2** untuk mendapatkan kondisi cube seperti di bawah ini:



Pada contoh ini, kamu mendapatkan cycle (UF-UL-UR-FR-DR). Misalkan pada saat eksekusi, kamu lupa di mana letak piece ketiga (UR). Lewati saja dengan menyelesaikan cycle (UF-UL-FR) sehingga menyisakan (UF-DR). Kemudian tiba-tiba di akhir eksekusi, kamu ingat bahwa piece ketiga letaknya di UR. Maka kembalilah pada piece tersebut dan ikut pertukarkan piece di urutan setelahnya (FR). Sehingga di akhir akan ada dua 2-cycle, yaitu (UF-DR)(UR-FR). Selesaikanlah dengan melakukan double-transposition.

Penjelasan step-by-step:

No.	Cycle	Keterangan
1		Awalnya cycle terdiri dari 5 stiker, yaitu <b>(UF-UL-UR-FR-DR)</b> . Sayangnya, kamu lupa di mana letak stiker di urutan ketiga (UL).
2		Kemudian loncati saja ke stiker di urutan setelahnya, yaitu FR, sehingga cycle-nya menjadi <b>(UF-UL-FR-DR)</b> .

3		Setup dengan gerakan <b>R</b> , lalu selesaikan <b>(UF-UL-FR)</b> dengan menggunakan <b>permutasi U CW</b> . Jangan lupa setelahnya undo gerakan dengan <b>R'</b> .
4		Tersisa <b>(UF-DR)</b> , dan ternyata kamu bisa ingat kembali stiker yang terlupakan, yaitu UR. Sehingga menjadi <b>(UF-DR)(UR-FR)</b> .
5		Lakukan setup <b>R' U'</b> untuk menyusun stiker menjadi Permutasi Z, lalu lakukan gerakan: <b>x - Permutasi Z - x'</b> . Undo setup move: <b>U R</b> .

## ◆ Algoritma Tambahan

Berikut ini adalah tambahan-tambahan algoritma yang bisa digunakan untuk menggantikan setup move yang sulit.

### 3 Corner di satu layer:

- (URF-ULB)(URB-DRB): **(R B' R' B)\*3**
- (URF-DRF)(URB-ULB): **(R U R' U - R U R' U2)\*2**
- Mirror dari kondisi di atas, (ULF-DLF)(URB-ULB): **(L' U' L U' - L' U' L' U2)\*2**

### 2 corner di layer U, 2 corners di layer D (diagonal)

- (UFR-DFR)(UBL-DBL): **B' - (R U R' U')\*3 - B atau B' - (U R U' R')\*3 - B**
- (UFR-UBL)(DFR-DBL): **L2 - x U2 - (H Permutation) - L2**

### 2 corner di layer U, 2 corners di layer D (bersebelahan)

- (UFL-UFR)(DFL-DFR): **D2 B2 - (E Permutation) - B2**
- (UFR-DBR)(UBR-DBR): **B - (R U R' U')\*3 - B'**
- (UFL-DBR)(UBL-DBR): **(R U' R' U)\*3 (R' U R U')\*3**

### Algoritma 5-Cycle (menukar 5 stiker edge sekali-gus)

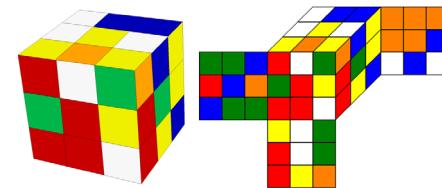
- (UF-UB-UR-UL-DB): **(R2 B2 R2 U)\*2**
- (UF-UL-UB-UR-DR): **(R2 U)\*6**

\* Notasi yang ditulis menggunakan format **(XX)\*n** misalnya seperti ini: **(RU)\*3**, berarti RU dilakukan berulang sampai 3 kali.

## ◆ Contoh Solving BLD

Acak cube kamu dengan warna merah di depan dan kuning di atas menggunakan algoritma ini:

**U' L2 F2 U F2 R2 D B2 F R' D' L2 R2 D L R' B' U2 R**



### Edge

- Alur stiker dari **UF** → **BD** → **LU** → **RD** → **UR** → **DL** → **FR** → **UF**
- Cycle I: **UF** → **BD** → **LU**  
        Setup: z LF'  
        U-Permutation CW  
        Undo setup: FL' z'
- Cycle II: **UF** → **RD** → **UR**  
        Setup: y F' R F  
        U-Permutation CW  
        Undo setup: F' R' F y'
- Cycle III: **UF** → **DL** → **FR**  
        Setup: z' L'R2  
        U-Permutation CW  
        Undo Setup: R2L z

- Alur stiker dari **UB** → LB → FL → RB → **UB**

Cycle I: UB → LB → FL

Setup: x' y' FLF'

U-Permutation CW

Undo setup: FLF' y x

Pada Cycle (ii) ini, didapatkan sisa edge yang berjumlah 2 buah yang saling bertukar posisi (2-cycle). Untuk menyelesaiakannya, kita harus menunggu setelah corner diselesaikan dimana nantinya pasti akan tersisa 2 corner. Lalu untuk menyelesaikan 2 edge dan 2 corner tersebut, kita dapat menggunakan algoritma PLL.

### Corner

- Didapatkan bahwa corner ULF sudah berada pada posisi yang benar (solved), maka kita abaikan saja dan mencari corner berikutnya.

- Alur stiker dari **URF** → URB → LUB → RDB → LDB → LDF → **RUF**

Diperoleh posisi start awal dan finish yang tidak sama, yaitu pada permukaan stiker yang berbeda. Hal ini berarti bahwa posisi corner URF nantinya akan terputar (twist).

Cycle I: URF → URB → LUB

Setup: L'UB' y'

A-Permutation CCW

Undo setup: y B U' L

Cycle II: URF → RDB → LDB

Setup: FLU2 z'

A-Permutation CCW

Undo setup: z U2L'F'

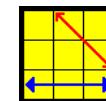
Pada Cycle (ii) ini, didapatkan sisa corner yang berjumlah 2 buah yang saling bertukar posisi (2-cycle). Untuk menyelesaiakannya, kita harus memasangkannya dengan 2 edge yang tersisa sewaktu solving edge tadi. Kemudian selesaikan dengan algoritma PLL.

### Parity

Dari solving edge dan corner, kita dapatkan kasus parity, yaitu 2 buah edge tertukar (**UB** ↔ RB) dan 2 buah corner tertukar (URF ↔ LDF). 2-cycle edge dan 2-cycle corner tersebut kita pasangkan menjadi kasus PLL.

Kamu bisa memilih dari 7 algoritma PLL yang menukar 2 edge dan 2 corner, yaitu permutasi T, V, F, R, J, Y, N. Untuk kasus ini, saya memilih permutasi R.

Setup movenya: r' D'L2 y', kemudian eksekusi permutasi R, lalu undo setup move: y L2 D r.



R U2 R D – R' U R D' – R' U' R' U – R U R' U

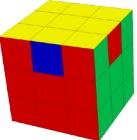
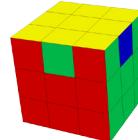
Stiker corner pada posisi FUR dan FDR terputar. Untuk membetulkan posisinya, gunakan algoritma untuk twist 2 corner: **x – R U2 - R' U' R U' R' - L' U2 - L U L' U L – x'**

## Metode TuRBO

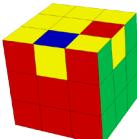
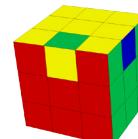
Metode ini dikembangkan oleh Erik Akkerdijk, yang sebenarnya juga merupakan metode berbasis cycle atau menukar piece/stiker dalam suatu siklus. Dengan TuRBO, setup move menjadi lebih mudah karena kita tinggal mengumpulkan piece-piece yang ingin ditukar di layer atas dan melakukan algoritma untuk menyelesaiakannya. Sehingga kamu yang telah cukup lancar menggunakan ABC bisa memperluas pengetahuan kamu dengan mempelajari algoritma-algoritma TuRBO.

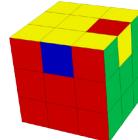
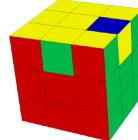
Berikut ini adalah list algoritma yang dipakai dalam TuRBO:

### ► EDGE

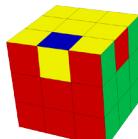
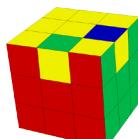
	
<b>(UF-UL-UR)</b> $M2 U' M U2 M' U' M2$ (Permutasi U CCW)	<b>(UF-UR-UL)</b> $M2 U M U2 M' U M2$ (Permutasi U CW)

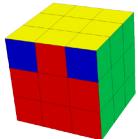
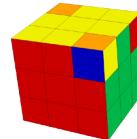
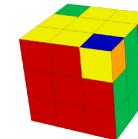
	
<b>(UF-LU-RU)</b> $M U M' U2 M U M'$	<b>(UF-RU-LU)</b> $M U' M' U2 M U' M'$

	
<b>(UF-UL-RU)</b> $L' U' L U M' U' L' U I U'$	<b>(UF-UR-LU)</b> $R U R' U' M' U R U' r' U$

	
<b>(UF-LU-UR)</b> $r U R' U' M U R U' R' U$	<b>(UF-RU-UL)</b> $I' U' L U M U' L' U L U'$

### ► CORNER

Clock Wise Cycle (searah jarum jam)		
		
<b>(UBR-UFR-UFL)</b> $(r' U r') \times D2 (L U' L')$ $D2 r2$	<b>(UBR-UFR-BLU)</b> $R' U2 R' D' R U2 R'$ $D R2$	<b>(UBR-UFR-LUB)</b> $U2 F' L' B' L F L' B$ $L U2$

(UBR-FRU-UBL) B L B' R B L' B' R' L2 F2 L B2 L' F2 L B2 L	(UBR-FRU-BLU) U' L' U R U' L U R'	

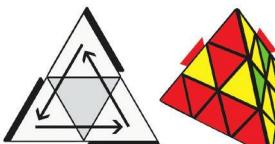
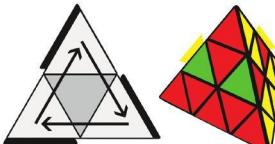
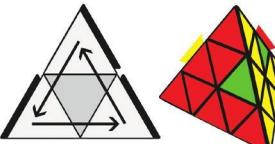
Counter Clock Wise Cycle (berlawanan arah jarum jam)		
(UBR-UBL-UFR) x R2 D2 R U R' D2 R U' R x'	(UBR-UBL-FRU) R B L B' R' B L' B'	(UBR-UBL-RUF) y R U2 R D R' U2 R D' R2 y'

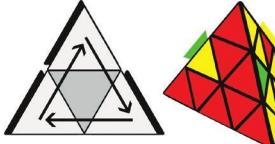
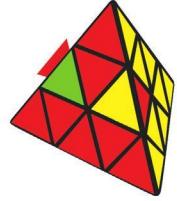
(UBR-BLU-UFR) R2 D' R U2 R' D R U2 R	(UBR-BLU-FRU) L' B2 L' F2 L B2 L' F2 L2	(UBR-BLU-RUF) R2 F' R2 F' L F R2 F' L' F2 R2

## Blindfold Pyraminx

Saya melakukan BLD pada Pyraminx karena ketertarikan saya dengan jenis *puzzle* satu ini. Ternyata metode BLD saya dengan cara cycling sticker ini bisa diterapkan pada Pyraminx, bahkan berbagai macam *twisty puzzle* lainnya, Megaminx misalnya.

Untuk kali ini, saya akan menjelaskan bagaimana cara melakukan BLD Pyraminx. Algoritma yang digunakan sama seperti algoritma-algoritma pada solving Pyraminx LBL, yaitu:

Kode	Pola	Algoritma
X1	 Cycle CCW	<b>R U R' U R U R'</b> atau <b>Rw' R L R' L R L R'</b> * Algoritma ini mirip dengan "Sune" pada 3x3
X2	 Cycle CW	<b>R' U' R U' R' U' R</b> atau <b>Rw' R L' R' L' R L' R'</b> * Inverse dari X1
Y1	 Cycle CCW sekaligus flip edge	<b>R' L R L' U' L' U L</b> atau <b>R' L R L' Lw R' L' R L</b> Cara pengenalan pola ini bisa dilihat dari adanya dua stiker yang letaknya saling berlawanan pada satu layer (stiker kuning) dan adanya dua stiker sewarna (merah).

Y2	 Cycle CW sekaligus flip edge	<b>L R' L' R U R U R'</b> atau <b>L R' L' R Rw' L R L R'</b> * Mirror dari X1
Z	 Flipping 2 edges	<b>L R' L' R U' R U R'</b> atau <b>L R' L' R Rw' L' R L R'</b> * Disebut juga algoritma "Flipper", yang akan membalik 2 edge.

### Contoh Solving Pyraminx BLD

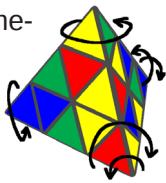
Acak Pyraminx kamu dengan warna merah di depan dan biru di bawah menggunakan algoritma ini:

**u r b' l u r' l' r b l u b r' u l**

#### ► Corner

- Atas: u'
- Kiri: **L Rw** – Cycle CCW – **Rw'** \*
- Kanan: **r'R Lw'** – Cycle CCW – **Lw**
- Belakang: **bB Rw'** – Cycle CCW – **Rw**

Untuk memorisasi, saya menggunakan metode visual yaitu membayangkan arah putaran dari keempat corner dan axial piece, seperti ilustrasi di samping. Tentu saja kamu bisa menggunakan metode memorisasi lainnya selain visual.



\* Notasi yang berwarna merah adalah setup dan undo setup move.

#### ► Edge

- Alur stiker dari **FL** → **LD** → **DR** → **LB** → **FD** → **LF**

**Cycle I:** **FL** → **LD** → **DR**

Setup: **R' U' L**

**Cycle CW**

Undo setup: **L U R**

**Cycle II:** **FL** → **LB** → **FD**

Setup: **R**

**Cycle CCW**

Undo setup: **R'**

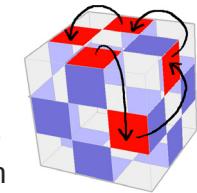
Kemudian untuk membalik (flipping) 2 edge, gunakan algoritma **LR'L'R – Rw' – L'RRLR'**.

## Memorisasi

Unsur paling penting selain memahami metode solving BLD adalah menggunakan metode memorisasi yang cocok. Untuk bisa solving BLD, tentunya kamu harus menghafal letak dan rute stiker pada cube terlebih dahulu. Ada berbagai metode memorisasi yang bisa diterapkan. Saya akan mencoba menjelaskan beberapa metode yang paling mudah dan banyak digunakan.

#### ► Visual

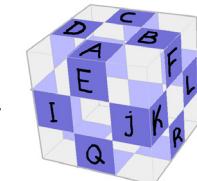
Merupakan metode memorisasi paling sederhana namun sulit untuk dilakukan, apalagi untuk pemula. Meskipun demikian, memorisasi visual dapat sangat cepat apabila dilatih. Beberapa BLD-ers top dunia menggunakan metode visual. Kelemahannya, hanya cocok untuk memorisasi jangka pendek karena sifatnya yang gampang dilupakan. Kurang cocok diterapkan untuk Multiple Blindfolded.



Menggunakan metode ini caranya cukup dengan membayangkan rute stiker secara visual. Bisa dibantu dengan tapping atau menyentuh stiker-stiker yang dimemo atau membayangkan stiker-stiker tersebut menyala seperti lampu ketika disentuh.

#### ► Alfabet atau Angka

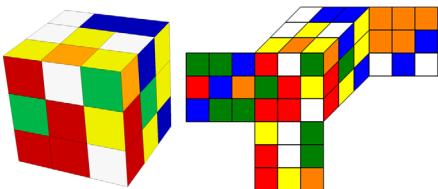
Memberi huruf untuk setiap stiker, seperti contoh gambar di samping ini. Misalkan dimulai dari stiker UF melingkar berlawanan arah jarum jam.



Bisa juga diganti dengan menggunakan angka untuk setiap stiker, dari satu sampai 24 untuk edge dan dari satu sampai 24 untuk corner.

Sebagai contoh, kita telaah kembali scramble pada contoh solving sebelumnya (bagian F). Acak cube kamu dengan warna merah di depan dan kuning di atas menggunakan algoritma ini:

**U' L2 F2 U F2 R2 D B2 F R' D' L2 R2 D L R' B' U2 R**



Pada contoh ini kita menemukan dua siklus (cycle) untuk edge, yaitu (UF-BD-LU-RD-UR-DL-FR) (UB-LB-FL-RB) dan satu siklus untuk corner (URF-URB-LUB-RDB-LDB- LDF).

Untuk solving edge, kali ini kita menggabungkan kedua cycle dengan memecah cycle kedua ke cycle pertama, sehingga alur cycle menjadi (UF-BD-LU-RD-UR-DL-FR-**UB**-LB-FL-RB-**UB**).

Kita akan menggunakan metode memorisasi yang paling sederhana, yaitu menggunakan alfabet atau huruf. Ingatlah huruf-huruf dari setiap stiker, sehingga item yang harus dimemorisasi menjadi (A-S-H-R-B-X-J-C-O-I-L-C). Kita bisa menghafal urutan huruf tersebut begitu saja, atau mengubah huruf-huruf tersebut menjadi kata-kata, misalnya Adi Suka Hemat Rajin Belajar, Xena Jalan-jalan Cari Obat, Indra Lemas Cacingan (ini hanya contoh☺).

#### ► PAO (Person-Action-Object)

Setiap stiker diberi nama orang, aktifitasnya, dan objek. Misalnya cycle stiker (UF-UR-FL). UF memiliki person "**Budi**", action "membaca", object "buku". UR memiliki person "**Peter**", action "**makan**", objectnya "spaghetti". Stiker FL memiliki person "**Andi**", action "bermain", objectnya "**kelereng**". Sehingga item memorisasi kita menjadi "**Budi-makan-kelereng**". Di pikiran, kita membayangkan Budi sedang makan kelereng yang besar-besar sampai tersedak (dengan memberi bayangan yang ekstrim atau dilebih-lebihkan, kita cenderung lebih mudah mengingat).

#### ► Image Based

Setiap stiker akan kita berikan suatu image tertentu untuk mewakilinya, image ini dapat berupa orang, benda atau suatu aktivitas. Misalnya cycle stiker (UF-UR-FL). UF memiliki image "**Donal Bebek**" UR memiliki image "**Paman Gober**". Stiker FL memiliki image "**Miki Tikus**". Sehingga item memorisasi kita menjadi "**Donal Bebek-Paman Gober-Miki Tikus**". Di pikiran, kita membayangkan Donal Bebek sedang minta uang kepada Paman Gober, sementara uangnya si Paman dicuri oleh Miki Tikus.

#### ► Journey Method (Metode Perjalanan)

Setiap stiker dilambangkan dengan suatu objek dan kita membayangkan di suatu tempat (sebaiknya tempat yang kita kenal baik, misalnya rumah kita). Kita mengambil suatu objek kemudian berjalan ke sudut lain dan mengambil objek berikutnya dan seterusnya.

Contohnya, cycle (UF-UR-FL). Stiker UF digantikan dengan objek “buku”, UR dengan objek “sisir”, FL dibayangkan dengan objek “piring”. Disarankan juga untuk memberi atribut yang berlebihan pada objek yang kita ingat, misalkan menambahkan ukuran, bau-bauan, warna, dan lain-lain. Pada saat memorisasi, kita membayangkan sedang berjalan di satu ruangan di rumah. Di atas meja kita mengambil **buku** yang sangat besar dan berat, kemudian berjalan ke sudut lain di ruangan dan menyisir rambut dengan **sisir** yang sangat besar, lalu berjalan lagi meraih **piring-piring** kotor di lantai dan melempar piring-piring tersebut hingga pecah berantakan.

#### ► Letter Pair

Metode ini pada dasarnya mirip dengan metode alfabet pada poin B, hanya pada aplikasinya setiap item diingat secara berpasang-pasangan dan diwakili oleh suatu image atau kata tertentu. Misalnya cycle (UF-UR-FL-RF) yang huruf-hurufnya adalah (A-B-I-K), diingat dengan memasangkan (pairing) huruf menjadi AB – IK. Misalnya AB mewakili “Abel” dan IK mewakili image “Ikan”. Di pikiran kita membayangkan “Abel sedang makan Ikan”. Chris Hardwick, pemegang rekor dunia BLD 5x5x5 menggunakan metode ini sebagai metode memorisasinya.

Demikianlah penjelasan mengenai metode blindfold yang saya gunakan. Mudah-mudahan dapat bermanfaat buat kamu. Jangan hanya terpaku pada rumusan yang saya berikan, tetapi cobalah untuk bereksperimen sebebas-

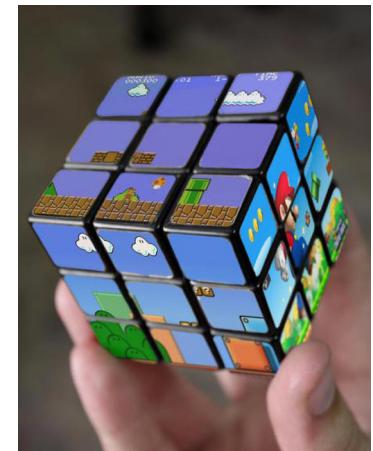
bebasnya dan mungkin kamu bisa menemukan *style* kamu sendiri dalam blindfold solving.

Untuk info-info tambahan dan update informasi terkait buku ini, kamu bisa mengakses website:

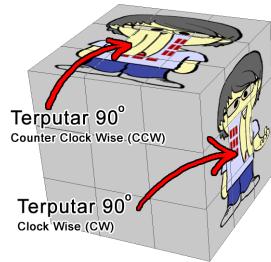
**www.mastercubing.com**

## Cara Menyelesaikan Cube 3x3x3 dengan Gambar/Logo

Rubik's Cube dengan gambar di permukaan sisi-nya lebih sulit untuk diselesaikan, karena kita harus menjaga gambar pada center piece agar tidak terputar atau terbalik. Padahal kebanyakan algoritma yang kita gunakan bisa mengubah arah orientasi center piece. Jangan kuatir... cobalah selesaikan Rubik's Cube tersebut dengan metode 3x3x3 biasa, lalu lihatlah kondisi center pada saat cube sudah ter-solved apakah sudah benar atau belum orientasinya. Jika belum, berikut ini saya berikan algoritma-algoritma untuk membetulkannya:



Pretty Patterns  
3x3x3

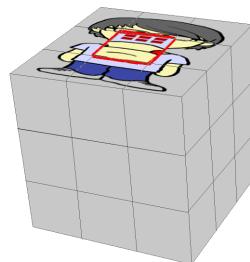


### Kondisi dua center terputar 90°

Algoritma untuk membetulkannya:

**U M E' M' - U' M E M'**.

Sedangkan inverse (kebalikan) dari algoritma tersebut akan memutar center U CCW dan center R CW:  
**M E' M' U - M E' M' U'**.



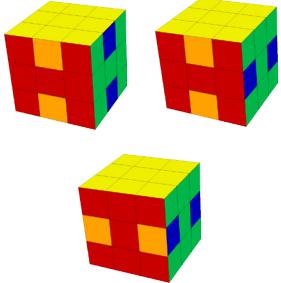
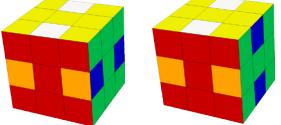
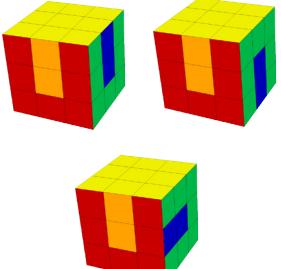
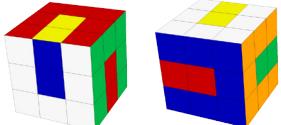
### Kondisi satu center terputar 180°

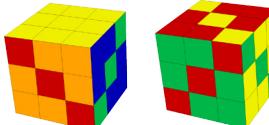
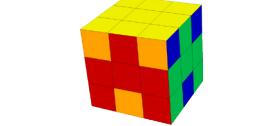
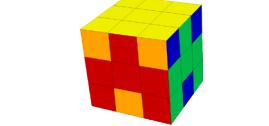
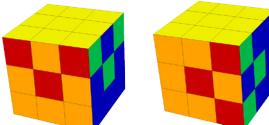
Algoritma untuk membetulkannya:

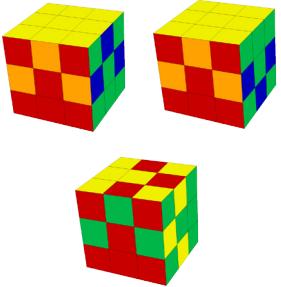
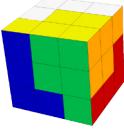
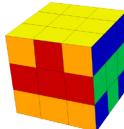
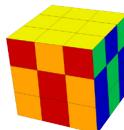
**U M E' M' - U' M E M'**.

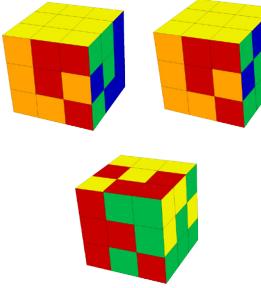
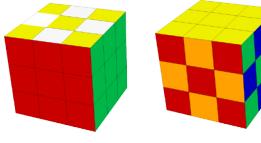
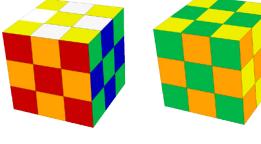
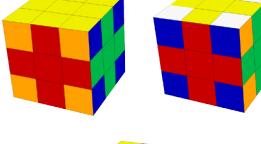
Setelah kamu bisa solving Rubik's Cube kamu, sekarang saatnya kamu mengetahui beberapa pola unik dan lucu yang bisa kamu bikin di Rubik's Cube 3x3x3 kamu. Nah, kalau sudah bisa bikin pola yang unik ini, Rubik's Cube-nya bisa kamu pajang deh. Berikut pretty patterns yang bisa kamu coba:

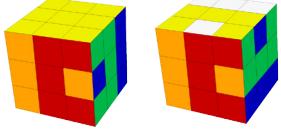
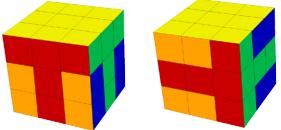
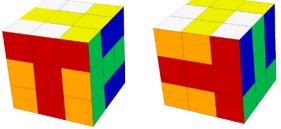
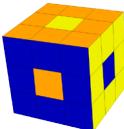
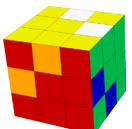
Nama	Pola	Algoritma
4 Dots		<ol style="list-style-type: none"> <li>(M' S) (M2' S')</li> <li>(M2' E')(M2' E)</li> </ol>
6 Dots		<ol style="list-style-type: none"> <li>M' S M S'</li> <li>R L' - U D' - F' B - R L'</li> </ol>
2 H		<ol style="list-style-type: none"> <li>M2' F2 M2' F2</li> <li>D2 - S M - D' M' D2 - S' D</li> </ol>

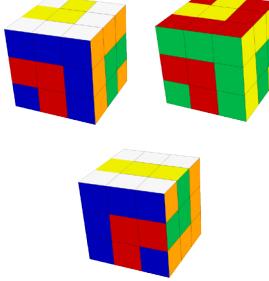
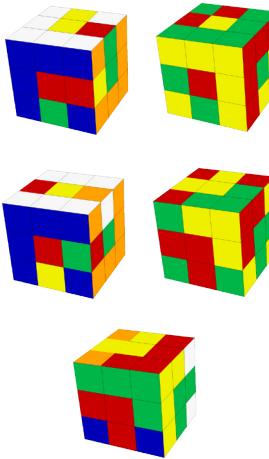
4 H		<ol style="list-style-type: none"> <li>1. (M2' S2 U2) (M2' S2 U2)</li> <li>2. R2 - U D - F2 B2 - U D - L2</li> <li>3. (S2 E) (S2 E)</li> </ol>
6 H		<ol style="list-style-type: none"> <li>1. E M2' E - F2 M2' F2</li> <li>2. (L2 B2 U2) (R2 L2) (U2 F2 L2)</li> </ol>
4 U		<ol style="list-style-type: none"> <li>1. (F' B') (D2 F B) (R' L') (D2 R L)</li> <li>2. U' (R L) (U2 D2) (R' L') U'</li> <li>3. U B2 U' - D' - R2 D2 L2 D</li> </ol>
6 U		<ol style="list-style-type: none"> <li>1. B' R F L B' - U D' - R B' L' B' D</li> <li>2. D2 f2 - D' L' U' L F' - D2 L' D L' F' D</li> </ol>

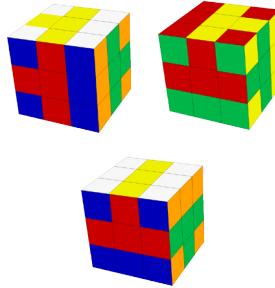
4 dan 6 Bars		<ol style="list-style-type: none"> <li>1. (F2 L2 B2) (F2 L2 B2)</li> <li>2. L2 D2 - R2 L2 - D2 L2</li> <li>3. U D - R2 U2 D2 - R2 D2</li> </ol>
Diag- onal		<ol style="list-style-type: none"> <li>1. U F2 R - S2 L2 S2 - R' F2 U D2</li> <li>2. D R' F L B2 L' - U D' - R L' - D2 F B' - D L2 F L</li> </ol>
4 A		M2' S2 - U2 - S2 M2'
4 D		<ol style="list-style-type: none"> <li>1. U2 - F B - D2 - F' B' R L - D2 - R' L'</li> <li>2. U2 F2 R - S2 L2 S2 R' - F2 D2</li> </ol>

4 dan 6 K		<ol style="list-style-type: none"> <li>1. U2 F2 U2 – F R2 – U2 D2 L2 – B D2 F2</li> <li>2. D R2 L2 D - U F2 B2 U</li> <li>3. L2 U2 R' B' – U' D – B2 D' R' D L D2 F U2 D L2</li> </ol>
6 L		R2 U2 L2 – F' B' – R2 D2 – F' B' – D2 L2 D2
4 Pertigaan		D M2' U2 M2' D
4 Y		B2 R2 F2 – B2 R2 F2 – U2

4 dan 6 Tanda Tanya		<ol style="list-style-type: none"> <li>1. U – R2 F M2' B2 M2' F' R2 – U' D2</li> <li>2. R L – D2 F2 D2 B2 – R L – B2 D2 F2</li> <li>3. U2 L' B' – U F D' – F' R F U – L' F2 – R' F' R F' R2</li> </ol>
2 dan 4 Pola Catur		<ol style="list-style-type: none"> <li>1. U2 – M2' E S2 E' – U2</li> <li>2. F2 M2' F2 – R2 S2 R2 – E2</li> </ol>
6 Pola Catur		<ol style="list-style-type: none"> <li>1. M2' S2 E2</li> <li>2. M' D M2' D2 – F2 B2 – U F' – E2 B2 M2' F' E2 M</li> </ol>
4 dan 6 Cross		<ol style="list-style-type: none"> <li>1. M2' U2 M2' – S2 D2 S2</li> <li>2. U S2 E2' R2 – S2 E2' L2 U'</li> <li>3. M S2 E D' S2 – U D – S2 U' F' E2 – F B – E2' B'</li> </ol>

4 dan 6 C		<ol style="list-style-type: none"> <li>1. R2 F E2' F2 E2' F' R2</li> <li>2. F' B' – L2 D2 – F B' – U2 R2 F2</li> </ol>
4 T		<ol style="list-style-type: none"> <li>1. U – R2 F2 B2 L2 – D – F2 R2 L2 B2</li> <li>2. R2 B R2 – U2 D2 – L2 F L2</li> </ol>
6 T		<ol style="list-style-type: none"> <li>1. F2 R2 U2 – F' B – D2 L2 – F B</li> <li>2. D2 R2 F2 – U D' – B2 L2 – U D</li> </ol>
3 Dots 1 Cin- cin		L' B' D' – L2 D F U B L2 – B' D' – R L' F
3 Ikan		D B2 R2 D' – R2 L2 – U2 D' – B2 R2 U'

Ular		<ol style="list-style-type: none"> <li>1. U2 D' B2 D R U R' – F' B2 U2 – L U' L F</li> <li>2. B R D' R' – F B' – D F' – U D' – L F L' U'</li> <li>3. U F M F' R2 F M' F – U' D – R2 F2 D'</li> </ol>
Ular War- na- warni		<ol style="list-style-type: none"> <li>1. F2 R2 – F B – R2 U2 B R2 – U D' – F2 L B2 U2</li> <li>2. U' B' R F – R L' – U' B – R L' – D' L' U B</li> <li>3. U – R L – F B – U D – R' L' – F B – D</li> <li>4. D2 L F B2 – U' R B U' – B2 D' R2 D – F B' – U F L' U'</li> <li>5. B' D F' L' – U F D' – B U' F – D' B' R D F' U</li> </ol>

Labi-rin		<ol style="list-style-type: none"> <li>1. R U' F2 U2 B' D' F2 B U' F L2 F' B' – D B R</li> <li>2. F' L2 D L' D' F2 L' F' D R D2 F R F' R2 D</li> <li>3. E2 D' L2 E2 R2 D</li> </ol>
----------	---	---

## Profil Speedcuber



**Muhammad Iril Khairul Anam**  
***"From Zero to Hero"***

Iril memulai petualangannya di dunia speedcubing sejak bulan April, setelah dia menonton tayangan acara *The Master* di sebuah stasiun televisi swasta. Ia tertarik setelah melihat penampilan Abel Brata di acara tersebut.

“Dulu waktu kecil pernah mencoba, tapi belum pernah berhasil. Dan di *The Master* itu pertama kali saya melihat orang bisa menyelesaikan kubus rubik,” tutur warga Lenteng Agung, Jakarta Selatan, ini. Terdorong penasaran dan rasa ingin bisa menyelesaikan Rubik’s Cube, ia pun men-

cari informasi tentang Rubik's Cube dan tutorialnya di internet. Ia pun pada bulan yang sama mulai bergabung dengan komunitas Jakarta Rubik's Cube Club (JRC).

Setelah beberapa bulan bermain, dia mulai tertarik dengan blindfolded cubing atau menyelesaikan kubus rubik dengan mata tertutup. Irlil pun mulai mempelajari dan berlatih metode BLD dan hasilnya dua minggu kemudian (26 Juli 2009), dia berhasil menjadi juara ketiga di kompetisi lokal yang diadakan Jakarta Rubik's Cube Club. Di bulan berikutnya, yaitu Agustus 2009, ketika diadakan kompetisi internasional Indonesia Open, Irlil kembali berhasil meraih medali perunggu di kategori Multiple Blindfolded. Hanya sekitar satu bulan semenjak pertama kali belajar!

Tak puas dengan catatan waktu ini, sejak awal Desember, ia berlatih keras mempersiapkan diri untuk ikut serta di kompetisi Jakarta Open pada akhir Januari lalu. Sahabat-sahabatnya di JRC pun ikut menemaninya berlatih setiap hari minggu di sekretariat klub. Di rumah, latihannya dilakukan dengan sembunyi-sembunyi karena orangtuanya, Abdurrahman dan Nurjanah, tidak ingin ia terlalu serius terjun ke dunia kubus Rubik. "Mereka takut saya melupakan kuliahan bila tiap hari berlatih Rubik," ungkapnya.

Berkat usaha kerasnya, Irlil berhasil memecahkan rekor dunia di kategori multiple blindfolded, pada Jakarta Open 2010, dengan menyelesaikan 16 buah Rubik's Cube dengan mata tertutup dalam waktu 56 menit 54 detik (termasuk memorisasi). Di kompetisi yang sama dia juga berhasil mencetak rekor Asia untuk kategori Multiple Blindfolded 5x5x5.

## Shotaro "Macky" Makisumi

Macky, begitu ia biasa dipanggil, memulai cubing sejak usia 9 tahun. Dia merupakan cuber pertama di dunia yang menembus AVG sub-15 dan menginspirasi banyak orang untuk mengikuti jejaknya. Dia sempat memegang rekor dunia untuk jangka waktu yang cukup lama, dari Januari 2004 sampai Oktober 2005. Tak cuma itu saja, dia juga memegang banyak rekor dunia di kategori lain. Tahun 2010, Macky mulai aktif kembali di dunia speedcubing dengan mengikuti beberapa kompetisi. Salah satunya Toulouse Open, akhir Januari 2010, dimana dia berhasil meraih juara pertama untuk kategori 3x3x3. *Welcome back, Macky!*

Selain di bidang cubing, Macky juga banyak memperoleh penghargaan dari berbagai kompetisi matematika yang dia ikuti. Dia terkadang masih menyempatkan diri menyapa penggemarnya di website miliknya, [www.cubefreak.net](http://www.cubefreak.net).



Informasi lainnya bisa kamu dapatkan di website resmi buku ini:

[www.mastercubing.com](http://www.mastercubing.com)

