



UNIVERSITAS INDONESIA

**MIGRASI *PRE STACK* DIMENSI KEDALAMAN UNTUK
PENCITRAAN STRUKTUR KOMPLEK DENGAN
ALGORITMA *LOCAL ANGLE DOMAIN***

TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister

KURNIAWAN ADHIPUTRA

0906576536

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM PASCA SARJANA GEOFISIKA RESERVOIR
UNIVERSITAS INDONESIA
JAKARTA
JULI 2011**

HALAMAN PERNYATAAN ORISINALITAS

**Tesis ini adalah hasil karya saya sendiri,
dan semua sumber baik dikutip maupun dirujuk
telah saya nyatakan dengan benar**

Nama : Kurniawan Adhiputra

NPM : 0906576536

Tanda Tangan :



Tanggal : 24 Juni 2011

HALAMAN PENGESAHAN

Tesis ini diajukan oleh :

Nama : Kurniawan Adhiputra

NPM : 0906576536

Program Studi : Geofisika Reservoir

Judul Tesis : Migrasi *Pre Stack* Dimensi Kedalaman Untuk Pencitraan Struktur Komplek Dengan Algoritma *Local Angle Domain*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Fisika pada Program Studi Geofisika Reservoir, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr. rer. nat. Abd. Haris ()

Ketua Sidang : Prof. Dr. Suprajitno Munadi ()

Penguji : Prof. Dr. Suprajitno Munadi ()

Penguji : Dr. Carlos Tarazona ()

Penguji : Dr. Charlie Wu ()

Ditetapkan di : Jakarta

Tanggal : 24 Juni 2011

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan tesis ini. Penulisan tesis ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Magister Program Studi Geofisika Reservoir, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tesis ini, sangatlah sulit bagi saya untuk menyelesaikan tesis ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Dr.rer.nat. Abdul, Haris, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
2. Pihak Seismic Geodata Processing, Geoscience Services Division, PT. Elnusa Tbk. yang telah banyak membantu terlaksananya penelitian ini;
3. Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
4. Sahabat yang telah banyak membantu saya dalam menyelesaikan tesis ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga tesis ini membawa manfaat bagi pengembangan ilmu.

Jakarta, Juli 2011

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Kurniawan Adhiputra
NPM : 0906576536
Program Studi : Geofisika Reservoir
Departemen : Fisika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Tesis

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Non eksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Migrasi *Pre Stack* Dimensi Kedalaman Untuk Pencitraan Struktur Komplek Dengan Algoritma *Local Angle Domain*,

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
Pada tanggal : Juli 2011
Yang menyatakan



(Kurniawan Adhiputra)

Abstract

Name : Kurniawan Adhiputra
Specialization : Reservoir geophysics
Title : Pre Stack Depth Migration for Imaging Complex Structure
With Local Angle Domain Algorithm

The structural interpretation of pre stack depth migrated seismic data has traditionally relied on full-offset stacks, attempting to take advantage of the stacking process to generate a high signal- to-noise ratio. However, in complex velocity environments (for example, subsalt), our observations suggest full-offset stacking can result in a lower signal-to-noise ratio than a partial stack of subsurface angle data.

Migration is a process which removes the effects of wave propagation from seismic data. Common Reflection Angle Migration (CRAM) is one of migration technique which is a multi-arrival, ray-based migration that uses the whole wavefield within a controlled aperture. Unlike conventional ray-based imaging methods, the ray tracing is performed from image points (in all directions, including turning rays) up to the surface, forming a system for mapping the recorded surface seismic data into the Local Angle Domain (LAD) at the image points.

The system enables extraction of high-resolution information about the subsurface model. Continuous structure surfaces, can be detected, even below complex geological structures. It is a target-oriented system, providing direct, high-resolution reservoir imaging, and high-resolution information in the vicinity of wells.

Keyword; Migration, raypath, angle, uniform

Abstrak

Name : Kurniawan Adhiputra
Peminatan : Geofisika Reservoir
Judul : Migrasi *Pre Stack* Dimensi Kedalaman untuk Pencitraan Struktur Komplek Dengan Algoritma *Local Angle Domain*

Interpretasi struktur dari data seismik *pre stack*, pada hasil migrasi kedalaman selalu bergantung kepada *stack full-offset*, dengan alasan adanya keuntungan peningkatan rasio S/N akibat proses *stacking*. Akan tetapi, pada area dengan struktur geologi yang kompleks (sebagai contoh, kubah garam), berdasarkan hasil pengamatan kita bisa menyimpulkan bahwa *stacking* dengan data *full-offset* bisa menghasilkan *image* yang kandungan rasio S/N lebih rendah dari hasil *partial stacking* data sudut *image* dibawah permukaan.

Migrasi adalah suatu proses yang berfungsi untuk menghilangkan efek dari penjalaran gelombang pada data seismic. *Common Reflection Angle Migration (CRAM)* merupakan salah satu tehnik migrasi yang berdasarkan konsep *multi-arrival*, migrasi yang bekerja berdasarkan *ray tracing*, menggunakan seluruh bagian gelombang dalam *aperture* yang terkontrol. Tidak seperti metode konvensional *ray tracing*, pada kasus ini *ray tracing* dilakukan dari titik *imaging* (dari segala arah, termasuk *turning rays*) naik menuju permukaan, membentuk suatu sistem yang berfungsi memetakan rekaman data seismik dipermukaan menjadi sesuatu yang dikenal sebagai *Local Angle Domain (LAD)* pada titik *imaging*. Prosedur ini bekerja berdasarkan iluminasi yang *uniform* dari segala arah pada titik *image*, memastikan bahwa semua sinar datang akan menjadi bahan perhitungan sementara keaslian amplitudo dan fase terjaga

Sistem ini memiliki kemampuan ekstraksi suatu informasi dengan resolusi yang baik mengenai model bawah permukaan. Objek yang memiliki struktur yang menerus hingga ke permukaan keberadaannya masih dapat terdeteksi, walaupun terletak dibawah struktur geologi yang kompleks. Karena berorientasi pada target, menyediakan secara langsung, *image* reservoir dengan resolusi tinggi dan informasi lainnya dalam batasan sumur

Kata kunci; Migrasi, *raypath*, sudut, *uniform*

DAFTAR ISI

Pernyataan Orisinalitas.....	ii
Lembar Pengesahan	iii
Kata Pengantar	iv
Lembar Penyataan Persetujuan Publikasi	v
Abstract	vi
Abstrak	vii
Daftar Isi	viii
Daftar Gambar.....	x
BAB 1. PENDAHULUAN	12
1.1 Latar Belakang	12
1.2 Tujuan Penelitian	13
1.3 Pembatasan Masalah	13
1.4 Metodologi Penelitian	13
1.5 Sistematika Penulisan	13
BAB 2. KONSEP DASAR MIGRASI	15
2.1 Definisi Migrasi.....	15
2.2 Model Reflektor Ledakan.....	17
2.3 Respon Impuls.....	20
2.4 Perbedaan Migrasi Waktu Dan Kedalaman.....	24
2.5 Migrasi <i>Pre Stack</i>	27
2.6 Migrasi <i>Pre Stack Kirchhoff</i>	28
2.7 Waktu tempuh.....	30
2.8 Migrasi <i>Pre Stack Kirchhoff Common Offset</i> dan Analisis Kecepatan Migrasi.....	33
2.9 Definisi Migrasi <i>CRAM</i>	35
2.9.1 Metode <i>CRAM</i>	36
2.9.2 Formula Migrasi <i>CRAM</i>	37
2.9.3 <i>KMAH Index</i>	39

BAB 3. APLIKASI ALGORITMA LOCAL ANGLE DOMAIN	41
3.1 Aplikasi Pada Data Sintetik.....	41
3.2 Aplikasi Pada Data Real.....	42
3.3 <i>Pre Stack Depth Migration</i>	44
3.3.1 Memperbaiki Model kecepatan Dengan <i>Tomography</i>	45
3.4 Final <i>PSDM</i>	46
BAB 4. HASIL DAN ANALISIS	48
4.1 Analisis Data Sintetik.....	48
4.2 Analisis Data Real.....	49
BAB 5. KESIMPULAN	53
DAFTAR ACUAN	54
LAMPIRAN	55

DAFTAR GAMBAR

Gambar 2.1. Simulasi penghilangan efek <i>mispositioning</i> karena reflector yang miring.....	16
Gambar 2.2. Model reflektor ledakan (<i>exploding reflector</i>).....	18
Gambar 2.3. <i>Stack section</i> sebelum dan setelah proses migrasi.....	18
Gambar 2.4. <i>Schematic</i> variasi kombinasi antara proses 2D dan 3D migrasi.....	19
Gambar 2.5. Penjumlahan kurva difraksi	21
Gambar 2.6. Geometri kurva difraksi hasil travel time.....	21
Gambar 2.7. <i>Reverse travel time</i>	22
Gambar 2.8. Superposisi impuls dan interferensi destruktif pada proses penjumlahan kurva difraksi.....	23
Gambar 2.9. Hasil migrasi dari superposisi respon impuls.....	23
Gambar 2.10. Perbedaan <i>raypath</i> antara penggunaan kecepatan <i>RMS</i> (gambar A) dan kecepatan interval (gambar B).....	25
Gambar 2.11. Permasalahan <i>lateral positioning</i>	26
Gambar 2.12. <i>Raypath</i> untuk <i>non zero offset</i>	29
Gambar 2.13. Permukaan kurva difraksi untuk offset yang dibatasi.....	29
Gambar 2.14. Data dari titik masukan yang disebarkan di <i>output space</i>	30
Gambar 2.15. 2 tipe sinar yang mungkin akan muncul.....	31
Gambar 2.16. 2 tipe sinar yang mungkin akan muncul-pada kasus kubah garam.....	32
Gambar 2.17. <i>Two steps</i> migrasi <i>Kirchhoff</i>	34
Gambar 2.18. Pengaplikasian kecepatan migrasi yang kurang tepat.....	35
Gambar 2.19. Contoh pasangan sinar (<i>incident</i> dan <i>scattered</i>) pada titik M dibawah permukaan dan 4 sudut yang berasosiasi dengan <i>Local Angle Domain</i>	37
Gambar 2.20. Kemungkinan yang dapat terjadi saat sinar memasuki segitiga.....	39
Gambar 3.1. Gambar <i>stack section</i> hasil migrasi dimensi kedalaman. (A). Hasil migrasi <i>CRAM</i> . (B). Hasil migrasi <i>Kirchhoff</i> ..	41

Gambar 3.2.	<i>Flowchart</i> penelitian.....	42
Gambar 3.3.	Model poligon $V_{interval}$ untuk dua lapisan teratas, menggunakan transformasi Dix.....	43
Gambar 3.4.	Proses <i>Coherency Inversion</i> untuk Horizon 3.....	43
Gambar 3.5.	V_{int} hasil transformasi Dix dan <i>Coherency Inversion</i>	44
Gambar 3.6.	Penampang seismik kedalaman hasil PSDM.....	45
Gambar 3.7.	<i>Model based tomography</i> pada horizon 6.....	46
Gambar 3.8.	V_{int} final hasil iterasi.....	47
Gambar 3.9.	<i>Stack section</i> final hasil migrasi.....	47
Gambar 3.10.	<i>Depth Migrated Gathers</i> hasil dari PSDM.....	47
Gambar 4.1.	CRP Gather hasil migrasi dimensi kedalaman. (A). Hasil migrasi <i>CRAM</i> . (B). Hasil migrasi <i>Kirchhoff</i>	48
Gambar 4.2.	Lintasan X1 Hasil PSDM dengan Algoritma <i>Kirchhoff</i> ..	49
Gambar 4.3.	Lintasan X1 Hasil PSDM dengan Algoritma <i>LAD</i>	49
Gambar 4.4.	Lintasan X2 Hasil PSDM dengan Algoritma <i>Kirchhoff</i> ..	50
Gambar 4.5.	Lintasan X2 Hasil PSDM dengan Algoritma <i>CRAM</i>	50
Gambar 4.6.	Lintasan X3 Hasil PSDM dengan Algoritma <i>Kirchhoff</i> (gambar disebelah kiri) dan algoritma <i>CRAM</i> (gambar disebelah kanan).....	51

BAB 1

PENDAHULUAN

1.1 Pendahuluan

Interpretasi struktur dari data seismik *pre stack*, hasil migrasi kedalaman selalu bergantung kepada stack *full-offset*, dengan alasan adanya keuntungan peningkatan rasio S/N akibat proses *stacking*. Akan tetapi, pada area dengan struktur geologi yang kompleks (sebagai contoh, kubah garam), berdasarkan hasil pengamatan kita bisa menyimpulkan bahwa *stacking* dengan data *full-offset* bisa menghasilkan *image* bawah permukaan yang kandungan rasio S/N lebih rendah dari hasil *partial stacking* data sudut. Walaupun menggunakan migrasi yang mengaplikasi penggunaan algoritma *multipathing* seperti PSDM dengan pendekatan persamaan gelombang ataupun CRAM, data seismic *full stack* secara umum, tergolong masih memiliki rasio S/N yang rendah, menyebabkan banyaknya interpretasi struktur yang bisa dikatakan “benar”.

Hal ini menyebabkan variasi masalah yang umumnya terjadi saat menentukan prospek, penempatan sumur, penentuan jebakan, volume dan perbedaan tingkat resiko.

Teori dan aplikasi metode *imaging* dengan menggunakan *Local Angle Domain* (LAD). (e.g., Miller et al., 1987, de Hoop and Bleistein, 1997, Brandsberg-Dahl et al. 1999, Rousseu et al. 2000, Xu et al., 2001, Audebert et al. 2002, Biondi and Symes, 2004, Ursin, 2004, Wu et al. 2006, Biondi, 2007).

Walaupun teori mengenai *imaging* di dimensi sudut sudah sangat dikenal, untuk pengaplikasiannya, terutama pada model 3D yang berukuran besar ataupun *imaging reservoir* dengan resolusi tinggi masih merupakan masalah yang penuh tantangan. Implementasi numerik untuk migrasi dimensi sudut berdasarkan amplitudo sinar sebenarnya pada data 3D yang sebenarnya dengan struktur geologi yang kompleks sudah pernah dilakukan oleh Koren et al. (2002). Tidak seperti metode *imaging* konvensional berdasarkan penjalaran sinar, *ray tracing* dilakukan dari titik *image* menuju ke permukaan dimana sinar difraksi satu arah akan ditelusuri ke segala arah termasuk sinar yang berbelok, membentuk system pasangan sinar untuk memetakan/pengelompokan rekaman data sismik

diper permukaan menjadi *gather* sudut refleksi. Pendekatan yang sama juga pernah dilakukan oleh *Brandsberg-Dahl et al.* (2003). Dan *Sollid and Ursin*(2003).

1.2 Tujuan Penelitian

Tujuan dari penelitian adalah untuk:

1. Mempelajari dan memahami konsep dasar dari algoritma *Local Angle Domain*
2. Mengaplikasikan algoritma *Local Angle Domain* pada data sintetik dan data *real*
3. Membandingkan hasil migrasi *pre stack depth migration* dengan menggunakan algoritma konvensional, dalam hal ini *kirchhoff*, dibandingkan dengan migrasi CRAM menggunakan pendekatan *Local Angle Domain*.

1.3 Pembatasan Masalah

Dalam penelitian ini, penulis menggunakan data set seismik 2D yang terdiri dari data set sintetik, meliputi *time gather* dan kecepatan *interval*. Untuk data real, terdiri dari 3 lintasan yang masing-masing meliputi *time gather*, kecepatan *RMS* dan informasi batas-batas formasi (horizon).

1.4 Metodologi Penelitian

Secara umum yang akan dilakukan adalah migrasi *pre stack depth migration* yang menggunakan data input *time gather* dan kecepatan *RMS*. Sehingga untuk melakukan migrasi kedalaman diperlukan beberapa langkah awal untuk persiapan data sebagai masukan input migrasi kedalaman, terutama membuat model kecepatan interval dari kecepatan *RMS*

1.5 Sistematika Penulisan

Penulisan skripsi ini terbagi menjadi lima bab yang saling berkaitan, yaitu pada **Bab 1 Pendahuluan**, bab ini berisi penjelasan secara singkat mengenai latar belakang, tujuan penelitian, batasan masalah serta metodologi penelitian. Kemudian dilanjutkan dengan **Bab 2 Dasar Teori**, bab ini berisi penjelasan teori

dasar dari migrasi. Pada **Bab 3 Aplikasi Metode *Local Angle Domain***, bab ini menjelaskan pengaplikasian pada data seismik baik pada data sintetik maupun pada data *real*

Bab 4 Hasil dan Analisis, bab ini berisi mengenai analisis terhadap hasil yang didapat dari bab sebelumnya. **Bab 5 Kesimpulan** berisikan kesimpulan dari penelitian.



BAB. 2

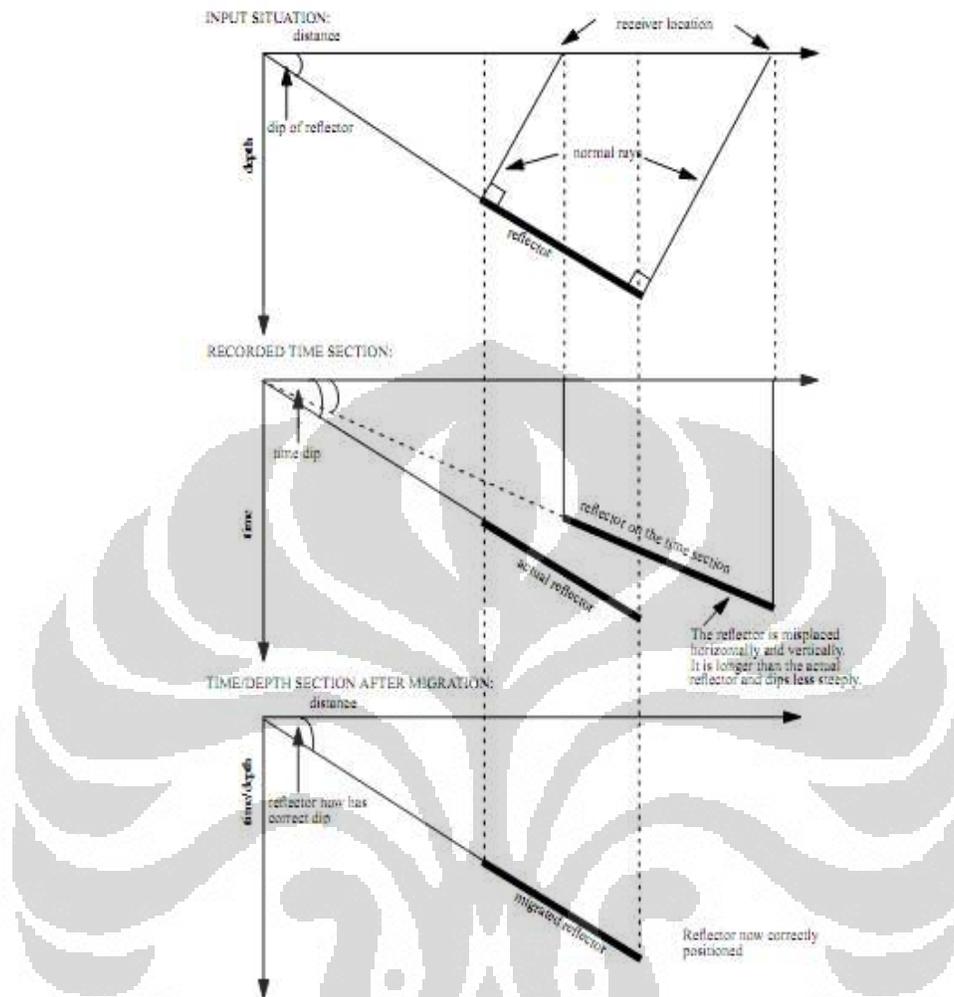
KONSEP DASAR MIGRASI

2.1 Definisi Migrasi

Migrasi adalah suatu proses yang berfungsi untuk menghilangkan efek dari penjalaran gelombang pada data seismik. Data seismik merupakan data hasil perekaman yang dihasilkan sebagai akibat dari penjalaran gelombang yang melalui bawah permukaan. Gambaran yang kita dapatkan melalui proses perekaman, mengalami distorsi sehingga tidak menunjukkan dengan benar gambaran sebenarnya dari geometri struktur bawah permukaan. Pada dimensi kedalaman, reflektor mendatar akan tetap muncul sebagai reflektor mendatar pada dimensi waktu, sementara untuk reflektor miring atau yang memiliki sudut kemiringan akan selalu berada pada posisi yang tidak sebenarnya pada gambaran seismik. Hal ini lah yang menjadi fungsi utama dari migrasi, yaitu untuk mengoreksi kesalahan posisi ini.

Sebagai contoh dapat dilihat pada **gambar 2.1**. Posisi sebenarnya dari suatu reflektor pada dimensi kedalaman akan berbeda dengan posisinya pada dimensi waktu. Tidak hanya posisi secara *lateral* dan *vertikal* saja, melainkan juga kemiringan dari reflektor tersebut. Pada proses penjalaran gelombang, proses ini ditandai oleh suatu sinar yang menjalar dari permukaan ke suatu reflektor dibawah permukaan dan sebaliknya. Sinar tersebut selalu digambarkan tegak lurus terhadap reflektornya. Pada dimensi waktu, *event – event* yang berasosiasi dengan *normal rays* digambarkan memiliki posisi *vertikal* pada posisi dimana sinarnya kembali ke permukaan (posisi penerima). Sehingga hal ini akan menyebabkan waktu tempuh semua reflektor miring akan lebih kecil dari pada yang sebenarnya dan reflektornya sendiri akan terlihat lebih panjang

Proses migrasi menghilangkan efek penjalaran gelombang dari rekaman data seismik dan sebagai hasilnya, event seismik akan berada pada posisi yang sebenarnya dibawah permukaan. Secara umum, migrasi akan membuat sudut kemiringan menjadi lebih curam, *event* miring menjadi lebih pendek, dan memindahkan *event updip*, sehingga antiklin akan mengecil dan sinklin akan bertambah ukurannya



Gambar 2.1. Simulasi penghilangan efek *mispositioning* karena reflector yang miring.
(Paradigm, 1999)

Sebagai tambahan dari distorsi geometri yang disebutkan diatas, Proses penjalaran gelombang juga mengakibatkan efek difraksi, sebagai akibat dari kehadiran ujung - ujung suatu reflektor sehingga menyebabkan perubahan amplitudo yang dikarenakan penyebaran dari muka gelombang. Migrasi memperbaiki efek - efek penjalaran gelombang diatas sekaligus memberikan beberapa keuntungan seperti :

- Migrasi *pre stack* memfasilitasi pemilihan kecepatan karena efek difraksi sudah dihilangkan, memfokuskan energi dan memperbaiki posisi *event* ke posisi yang sebenarnya.

- Migrasi *pre stack* dapat digunakan untuk analisis AVO, jika migrasi dilakukan dengan memperhatikan faktor *preserve amplitude*
- Migrasi meningkatkan resolusi *lateral*.
- Migrasi mengurangi *random noise*.

Proses migrasi juga dapat didefinisikan dengan persamaan gelombang dan pada umumnya berdasarkan persamaan penjalaran gelombang akustik 1 arah. Maka dari itu, persamaan ini tidak dapat menghilangkan pengaruh gelombang *shear*. Selain itu juga didasarkan penggunaan asumsi bahwa media penjalarnya isotropi, juga tidak dapat mengatasi *noise multiple* dengan baik.

Secara sederhana, migrasi dapat dikatakan sebagai suatu proses yang seolah - olah memutarbalikan penjalaran gelombang. Aspek-aspek tehnikal dari migrasi dapat dilihat dari beberapa sudut pandang , yaitu :

- Migrasi sebagai proses penjalaran kontinyu ke bawah permukaan (*downward continuity*)
- Migrasi sebagai proses penjumlahan difraksi (*diffraction summation*)
- Migrasi menggunakan prosedur waktu terbalik. (*reverse time*)

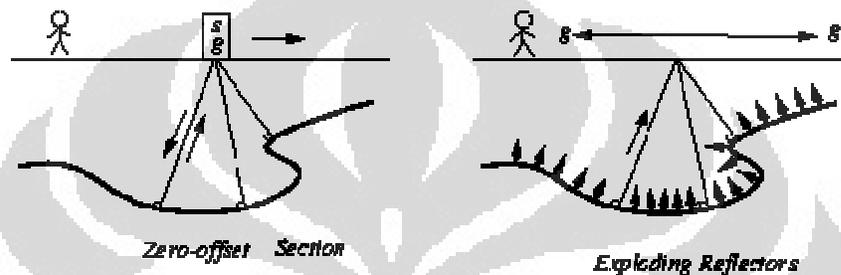
Ketiganya merupakan prosedur yang relatif sama, hanya menggunakan pendekatan yang berbeda dari yang umum digunakan.

2.2 Model Reflektor Ledakan

Suatu model dasar dibutuhkan untuk mendefinisikan migrasi pada kasus *zero offset*, yaitu model reflektor ledakan (*exploding reflector*). Model ini dapat memberikan gambaran untuk mensimulasikan seismik *zero offset*

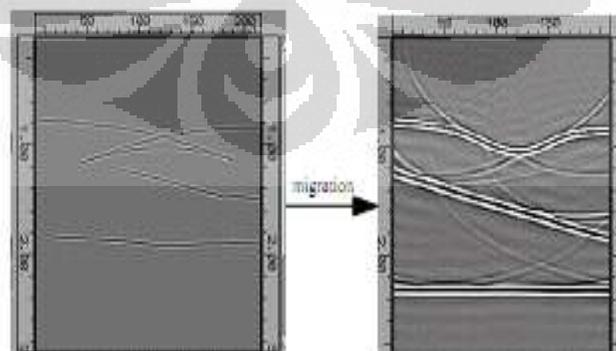
Jika kita asumsikan bahwa semua reflektor seismik dibawah permukaan tersusun dari titik-titik yang berdekatan. Setiap titik berfungsi sebagai sumber ledakan pada saat waktu awal ($t=0$), dimana semua sumber meledak secara bersamaan, memulai proses penjalaran gelombang. Penerima yang berada dipermukaan merekam waktu tempuh saat gelombang tiba dipermukaan. Dengan menggunakan konfigurasi ini, hasil yang didapat menunjukkan bahwa rekaman

data merupakan simulasi dari *stacked section*, jika model kecepatan yang digunakan setengah dari kecepatan yang sebenarnya. Kecepatan yang digunakan pada model ini sebaiknya memiliki nilai setengah dari nilai sebenarnya dikarenakan yang digunakan adalah waktu tempuh satu arah dari titik bawah permukaan menuju ke permukaan, sebanding dengan jika menggunakan waktu tempuh bolak-balik dari permukaan menuju ke reflektor dan kemudian kembali ke permukaan.



Gambar 2.2. Model reflektor ledakan (*exploding reflector*). (sepwww.stanford.edu, 2011)

Model reflektor ledakan menggambarkan proses penjalaran gelombang yang dapat menghasilkan *section zero offset* dalam satu eksperimen tunggal. Berdasarkan model di atas, migrasi dapat dijelaskan sebagai suatu prosedur yang memutar balik proses penjalaran gelombang, dari suatu reflektor kembali ke titik awal disaat $t=0$. Input data yang digunakan untuk migrasi adalah *section* waktu yang direkam dipermukaan. Hasil dari migrasi merupakan *section* seismik yang dapat mensimulasikan situasi awal sebelum proses penjalaran gelombang dimulai.

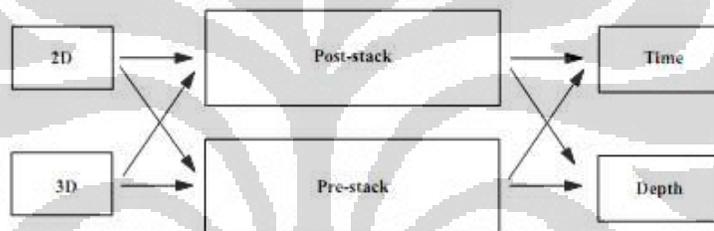


Gambar 2.3. *Stack section* sebelum dan setelah proses migrasi.

Mengacu pada model diatas, pada **gambar 2.2**, migrasi (*post stack*) secara umum melibatkan 2 tahapan :

- Menjalankan proses penjalaran gelombang yang berjalan mundur
- Memilih bagian dari gelombang yang tiba saat waktu *zero*. Tahapan ini dinamakan *imaging*, dan syarat $t=0$ disebut *imaging condition*

Variasi dari migrasi dapat diklasifikasikan menjadi 2; berdasarkan pada dimensi dimana migrasi dilakukan atau berdasarkan pada algoritma yang digunakan. Klasifikasi berdasarkan dimensi membedakan antara migrasi 2D dan 3D, *post* dan *pre stack* migrasi serta migrasi dimensi waktu atau dimensi kedalaman. Kemungkinan untuk mengkombinasiakan diantara semua sangat dimungkinkan;



Gambar 2.4. Schematic variasi kombinasi antara proses 2D dan 3D migrasi.

Pengklasifikasian yang lain adalah klasifikasi berdasarkan algoritma yang digunakan. Algoritma yang banyak digunakan termasuk diantaranya :

- F-K
- *Kirchhoff*
- *Phase Shift*
- PSPC (*Split Step*)
- F-X
- *Finite Difference*

Salah satu alasan utama yang memunculkan variasi migrasi, dikarenakan proses ini termasuk proses yang relatif mahal harganya. Maka dari itu pemilihan migrasi yang digunakan akan sangat bergantung kepada kompleksitas dari data yang digunakan. Walaupun terlihat sangat ideal, satu jenis migrasi cukup untuk digunakan dalam mengatasi semua kemungkinan permasalahan yang muncul,

disebabkan dari segi harga, program migrasi yang lain dipilih cenderung menyesuaikan dengan situasi.

Algoritma migrasi yang terbaik sebaiknya memenuhi beberapa kriteria :

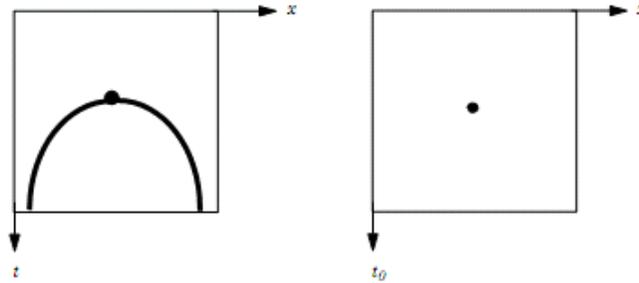
- Tidak terlalu mahal
- Dapat mengatasi kemiringan yang curam
- mengatasi variasi perubahan kecepatan secara *lateral*
- *Preserve amplitude* dan fase
- Menghasilkan artifak (*noise*) yang minimum
- Tidak terlalu sensitif terhadap kualitas data (rasio S/N cukup rendah) dan kecepatan model

Perbedaan utama dari variasi program migrasi adalah sejauh mana mereka dapat memenuhi kriteria diatas. Sebagai contoh, dimana waktu komputasi yang digunakan dalam proses migrasi data *pre stack* jauh lebih banyak dibandingkan migrasi data *post stack*, sehingga proses migrasi data *pre stack* lebih mahal dan membutuhkan peralatan yang memadai. Sehingga diwaktu lampau migrasi *pre stack* hanya digunakan pada data yang memiliki struktur kompleks. Namun pada saat ini, diiringi dengan kemajuan teknologi, migrasi *pre stack* sudah menjadi prosedur standar. Pada kasus data 3D data, waktu komputasi yang diperlukan juga jauh lebih banyak dari kasus 2D, sehingga migrasi 3D tetap menjadi proses yang rumit dan mahal.

Salah satu point yang penting adalah pada migrasi *pre stack* biasanya dilakukan dalam beberapa iterasi untuk membuat kecepatan model final. sehingga, algoritma yang paling cepat digunakan saat tahapan awal dimana hanya pada saat tahap akhir saja migrasi yang lebih akurat dan mahal digunakan untuk mendapatkan hasil akhirnya.

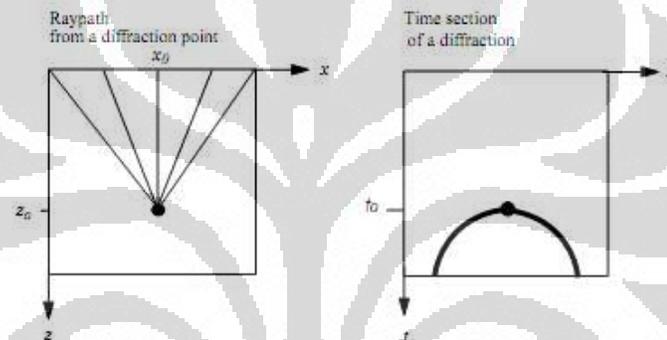
2.3 Respon Impuls

Respon impuls sering digunakan sebagai suatu test yang dianggap cukup mudah dan dapat mewakili untuk mengevaluasi proses migrasi. Untuk memahaminya kita perlu memulai dari proses penjumlahan kurva difraksi. Penjumlahan kurva difraksi dilakukan dengan menjumlahkan seluruh kurva difraksi dan meletakkan hasilnya pada puncak dari hiperbolanya



Gambar 2.5. Penjumlahan kurva difraksi

Geometri dari kurva difraksi dibentuk oleh waktu tempuh *ray* dari titik difraksi, yang terletak di bawah permukaan menuju seluruh titik di permukaan.



Gambar 2.6. Geometri kurva difraksi hasil travel time. (Paradigm, 1999)

Persamaan pembentuk kurva difraksi:

$$t = \sqrt{t_0^2 + \frac{2(x-x_0)^2}{v^2}} \quad (2.1)$$

Dimana (x_0, z_0) merupakan titik difraksi dan $t_0 = \frac{2-z_0}{v_0}$

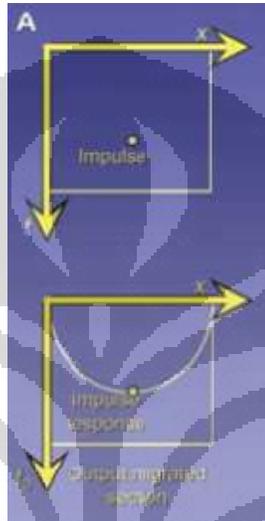
Persamaan diatas mendefinisikan penjumlahan difraksi untuk setiap titik (x_0, z_0) pada *section* hasil migrasi. Semua titik (x, t) pada *input section* yang mengikuti hubungan diatas akan dijumlahkan dan hasilnya akan diposisikan pada titik (x_0, z_0)

Dengan mengubah sudut pandang kita juga bisa mendapatkan

$$t_0 = \sqrt{t^2 + \frac{2(x-x_0)^2}{v^2}} \quad (2.2)$$

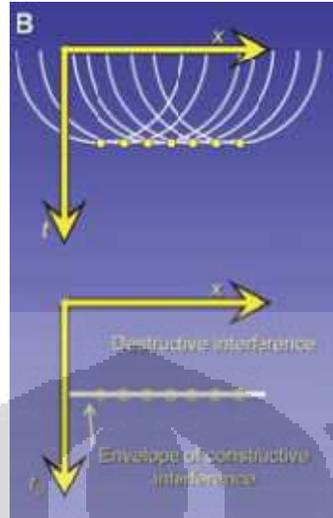
Hasil dari migrasi suatu titik input disebut impuls respon dari suatu migrasi. Impuls respon merupakan suatu tempat dengan geometri tertentu pada

hasil migrasi dimana titik input dipetakan. Hal tersebut merepresentasikan semua kemungkinan titik kedalaman yang dapat membuat sinyal pada *input point* (x, t) pada *section* waktu. Jalan lain untuk menginterpretasikan impuls respon adalah dengan menggunakan model bawah permukaan yang hanya menghasilkan satu titik pada *section* waktu. Dimana model ini harus berbentuk kurva (ellips) yang mengkonsentrasikan semua energi ke satu titik, seperti gambar dibawah ini.



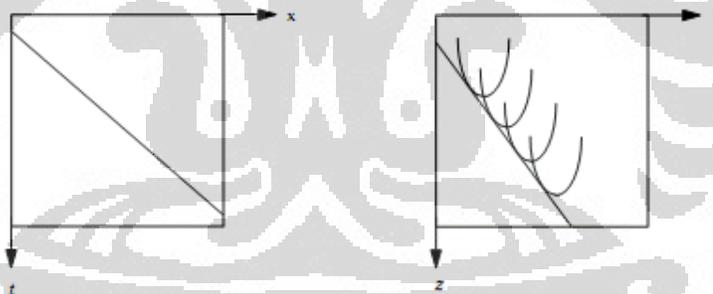
Gambar 2.7. *Reverse travel time.* (Helsing, C. Edward, et al. 2007)

Migrasi dengan algoritma penjumlahan kurva difraksi dilakukan dengan menjumlahkan amplitudo sepanjang kurva difraksi pada data input atau dengan kata lain, menyebarkan amplitudo sepanjang kurva ellips pada hasil migrasi. Kedua prosedur ini faktanya sangat identik. Dengan pendekatan kedua, setiap titik masukan dimigrasi secara terpisah, dan hasilnya merupakan superposisi dari setiap impuls. Pencitraan yang terbentuk merupakan hasil pada garis singgung (*envelope*) semua kurva elliptic hasil interferensi. Ujung-ujung dari kurva terhapuskan sebagai akibat dari *interferensi destruktif*.



Gambar 2.8. Superposisi impuls dan interferensi destruktif pada proses penjumlahan kurva difraksi. (Helsing, C. Edward, et al. 2007)

Saat terdapat reflektor mendatar, bagian dari kurva ellips yang berkontribusi pada penciptaan gambaran adalah bagian titik paling bawah. Untuk reflektor yang memiliki kemiringan, bagian lereng adalah bagian yang berkontribusi untuk membentuk gambaran.



Gambar 2.9. Hasil migrasi dari superposisi respon impuls

Semakin curam kemiringannya, semakin banyak bagian dari ellips yang ikut berkontribusi. Maka dari itu, dengan memeriksa respon impuls, kita dapat mengetahui respon kemiringan saat melakukan migrasi. Jika bagian ujung dari ellips ada yang terpotong, artinya adalah proses migrasi dibatasi oleh kemiringan dan kemiringan yang sangat curam tidak akan tergambarkan.

2.4 Perbedaan migrasi waktu dan kedalaman

Suatu kesalahan umum untuk sebagian orang adalah anggapan, bahwa migrasi waktu bekerja pada dimensi waktu dan migrasi kedalaman bekerja di dimensi kedalaman. Sebenarnya, semua migrasi diformulasikan pada dimensi kedalaman. Migrasi didefinisikan sebagai suatu transformasi dari dimensi waktu $f(x, t, z = 0)$ menjadi dimensi kedalaman $(x, t = 0, z)$. Perbedaan utama antara migrasi waktu dan kedalaman adalah bagaimana cara mereka menangani perubahan kecepatan kearah lateral. Migrasi waktu tidak dapat mengatasi variasi perubahan kecepatan kearah lateral dengan baik, sementara migrasi kedalaman dapat mengatasi hal ini. Hal yang harus diperhatikan bahwa keluaran hasil migrasi baik waktu maupun kedalaman bukan merupakan suatu hal yang fundamental. Pada sebagian besar kasus, migrasi waktu mengasumsikan bahwa model kecepatan dapat merepresentasikan (secara lokal) *gradient* kecepatan *vertikal*. Sehingga perbedaan antara sumbu waktu dan kedalaman hanya permasalahan *scaling*. Hal ini yang menjadi alasan kenapa hasil dari migrasi waktu dapat ditampilkan dalam dimensi waktu. Lebih dari itu, untuk migrasi waktu, kemenerusan penjalaran gelombang ke bawah dapat diproses dalam tahapan Δt yang tetap.

Terjaga untuk setiap *sample*, dapat dituliskan hubungannya :

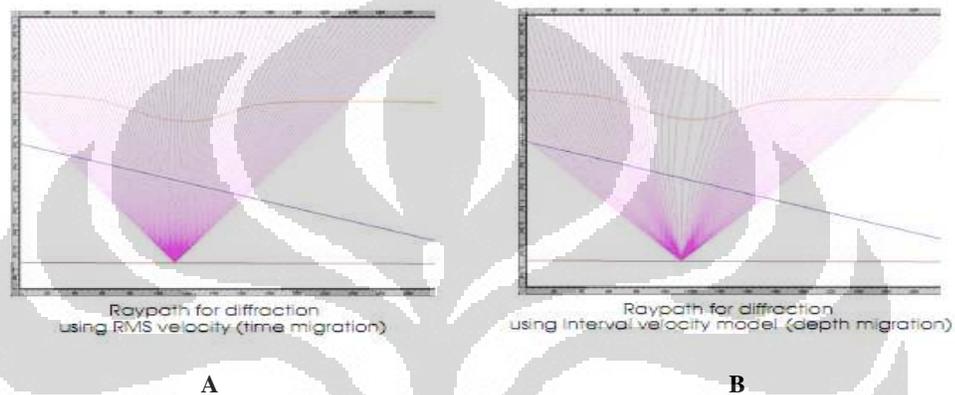
$$\Delta z = \Delta t \cdot \frac{v}{2} \quad (2.3)$$

Δt merupakan *sampling* waktu yang konstan dari *section* waktu. Dengan pendekatan ini, hasil dari migrasi waktu dapat ditampilkan secara langsung sebagai *section* waktu dikarenakan *sampling* waktu Δt .

Pada akhirnya, ada perbedaan yang sangat fundamental antara migrasi waktu dan kedalaman. Hal ini dapat dijelaskan dengan menggunakan migrasi yang mengaplikasikan algoritma penjumlahan kurva difraksi. Dengan migrasi penjumlahan kurva difraksi, nilai - nilai dari data dijumlahkan sepanjang hiperbola difraksi dan hasilnya diletakkan pada puncaknya. Prosedur ini, saat dilakukan pada migrasi waktu, menggunakan kecepatan *RMS* pada setiap titik dan mengasumsikan bahwa kurva difraksi berupa hiperbolik

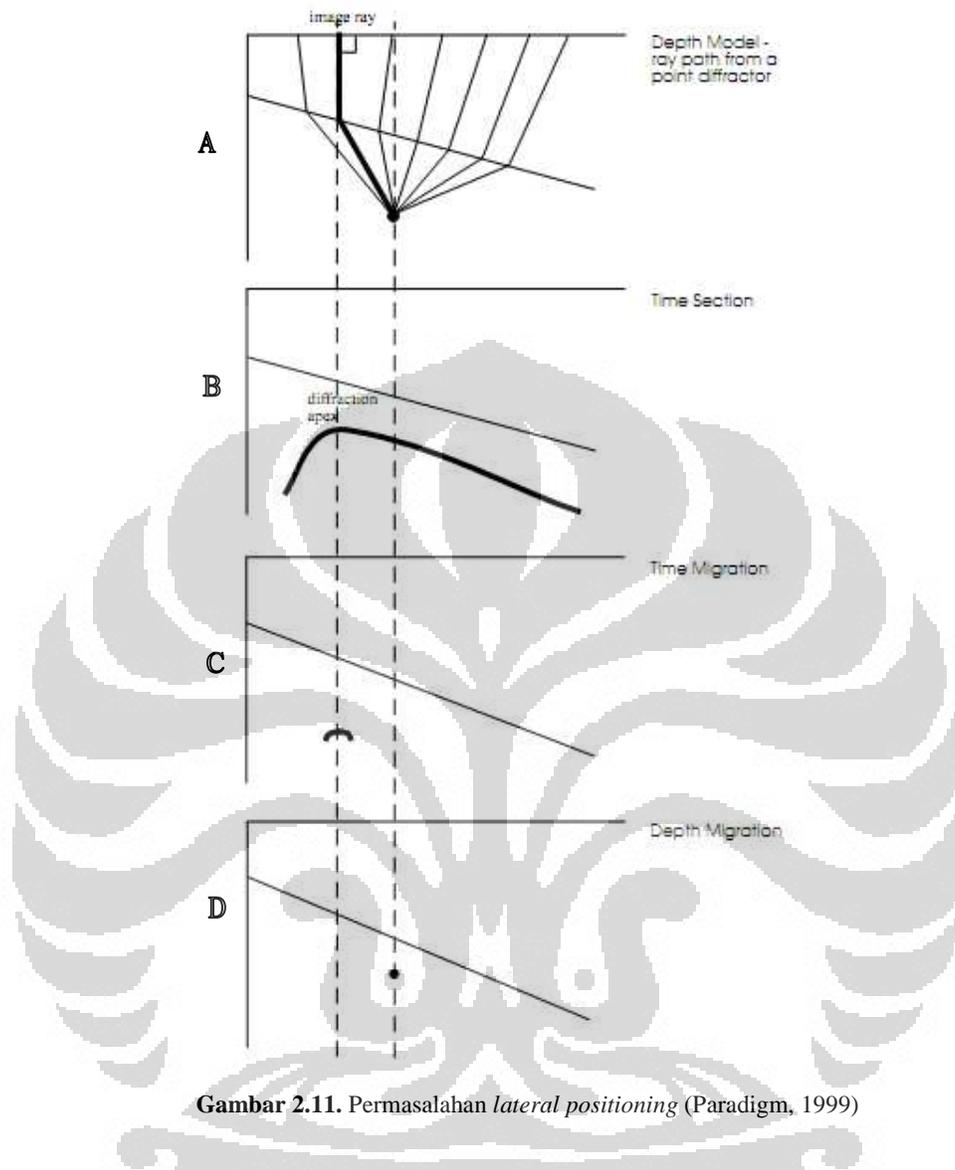
$$t = \sqrt{t_0^2 + \frac{2(x-x_0)^2}{v_{RMS}^2}} \quad (2.4)$$

Dimana t_0 merupakan waktu migrasi dan x_0 merupakan titik difraksi. Sedangkan migrasi kedalaman tidak bergantung pada asumsi hiperbolik. Migrasi kedalaman menggunakan model kecepatan interval untuk melakukan penelusuran sinar dari titik difraksi menuju ke permukaan, memastikan bahwa setiap kurva difraksi pada setiap titik dihitung dengan akurat. Migrasi akan melakukan perhitungan penjumlahan nilai data sepanjang kurva difraksi dan meletakkannya pada titik difraksi



Gambar 2.10. Perbedaan *raypath* antara penggunaan kecepatan *RMS* (gambar A) dan kecepatan interval (gambar B).

Saat model kecepatan bervariasi secara lateral, seperti contoh **gambar 2.10**, maka model kecepatan *RMS* gagal untuk mendeskripsikan geometri dari difraksi. Dalam hal ini, antara migrasi waktu dan kedalaman tidak memberikan hasil yang sama. Mereka berbeda dalam hal kualitas gambar dan posisi secara lateral. Hasil yang lebih tepat didapatkan melalui migrasi kedalaman, dimana penjumlahan energi dilakukan sepanjang kurva yang sebenarnya bukan menggunakan kurva hasil pendekatan. Sehingga, gambar hasil migrasi yang didapatkan lebih baik dengan migrasi kedalaman dibanding dengan migrasi waktu (diasumsikan model kecepatan interval yang digunakan sudah benar)



Gambar 2.11. Permasalahan *lateral positioning* (Paradigm, 1999)

Masalah *positioning lateral* diilustrasikan seperti **gambar 2.11**. Gambar 2.11 bagian A merupakan model yang digunakan. Saat kecepatan bervariasi secara lateral, puncak dari titik difraksi (waktu minimum) tidak berhubungan langsung dengan titik difraksi (gambar 2.11 bagian B). Migrasi waktu memposisikan *image* pada puncak (waktu minimum), sehingga secara *lateral* terjadi kesalahan posisi event (gambar 2.11 bagian C). Sebaliknya migrasi kedalaman memposisikan event dengan benar (gambar 2.11 bagian D).

Hal ini terjadi karena puncak dari difraksi terletak ditempat dimana *image rays* muncul dipermukaan. *Image ray* (garis tebal pada gambar diatas) merupakan

sinar yang berjalan dari permukaan menuju titik difraksi sehingga sinarnya membentuk normal *incidence* terhadap permukaan. Ini merupakan sinar tercepat antara permukaan dengan titik difraksi dan sinar ini mengidentifikasi puncak dari kurva difraksi.

Hal yang perlu diingat, dalam contoh diatas pada **gambar 2.11**, tidak semua migrasi waktu akan menghasilkan jawaban yang sama. Diwajibkan untuk mendefinisikan perbedaan antara migrasi waktu dan kedalaman menggunakan *image rays* dan penjumlahan kurva difraksi. Akan tetapi, beberapa migrasi yang tidak berdasarkan penjumlahan kurva difraksi, seperti metode *downward continuation*, mungkin berdasarkan prinsip kinematik gelombang yang berbeda.

Migrasi waktu diketahui sebagai sesuatu yang sangat umum (*robust*), dan kurang sensitif terhadap model kecepatan dibanding dengan migrasi kedalaman. Saat model kecepatan yang digunakan salah, migrasi kedalaman akan melakukan perhitungan geometri yang tepat dari kurva difraksi sehingga dapat menghasilkan kurva yang sangat berbeda dari kurva sebenarnya. Hal ini akan menghasilkan *image* yang sangat rendah kualitasnya.

Dapat ditarik kesimpulan, posisi *lateral* dari suatu *event* hasil migrasi waktu bisa memiliki perbedaan posisi *lateral* dengan hasil migrasi kedalaman.

2.5 *Pre stack Migration*

Untuk struktur yang tidak kompleks, hasil refleksi memiliki kurva hiperbolik dan proses penjumlahan (*stack*) berjalan dengan baik. Untuk struktur yang sebaliknya, memiliki kompleksitas, proses penjumlahan bisa mengakibatkan pengurangan data-data yang berharga. Sayangnya, dalam hal ini migrasi sangat diperlukan. Sehingga sangat menguntungkan untuk melakukan migrasi sebelum melakukan *stacking*. Dengan melakukan hal itu, energi akan terfokuskan sebelum *stacking* dan data masukan *stack* akan terlihat lebih mudah dimengerti. Selain itu juga, *event* akan berada pada posisi yang benar sebelum proses *stack* dimana hal ini membantu proses *stack* itu sendiri. Migrasi *pre stack*, saat dilakukan sebagai migrasi kedalaman, dapat mengatasi *moveout* yang tidak memenuhi kurva hiperbolik; hal ini secara signifikan meningkatkan kualitas hasil

Migrasi *pre stack* dilakukan sebelum proses NMO dan *stack*. Proses ini mengoreksi *lateral miss positioning* dari *event* refleksi sekaligus melakukan proses *move out*, karena migrasi mengaplikasikan koreksi *move out*. Migrasi *pre stack* sangat sensitif terhadap akurasi model kecepatan. Walaupun tidak menguntungkan, sensitivitas migrasi *pre stack* terhadap model kecepatannya dapat digunakan untuk melakukan analisis kecepatan. Hal ini menjadikan migrasi waktu *pre stack* sebagai suatu alat untuk mendapatkan kecepatan *RMS* yang sebenarnya, dan migrasi kedalaman *pre stack* menjadi alat untuk mendapatkan kecepatan interval.

Analisis kecepatan migrasi dilakukan pada posisi migrasi yang sudah terkoreksi posisinya, mengakibatkan peningkatan rasio S/N karena energinya yang sudah terfokuskan. Analisis kecepatan setelah migrasi tidak *terinterferensi* oleh energi difraksi yang sudah dihilangkan saat migrasi. Sebagai tambahan, kecepatan didapatkan tanpa menggunakan pengasumsian *move out* yang digunakan hiperbolik

2.6 Migrasi *Pre stack Kirchhoff*

Migrasi *Kirchhoff* dapat dengan mudah dimodifikasi menjadi migrasi *pre stack*. Migrasi *Kirchhoff* pada dasarnya merupakan penjumlahan kurva difraksi. Jika kurva difraksi diketahui, maka penjumlahan dapat dilakukan baik secara *pre stack* maupun *post stack*. Pada kasus *zero offset*, kurva difraksi berupa hiperbolik dan dapat didefinisikan sebagai berikut :

$$t = \sqrt{t_0^2 + \frac{(x-x_0)^2}{v^2}} \quad (2.5)$$

Dimana (x_0, z_0) merupakan lokasi titik difraksi dan

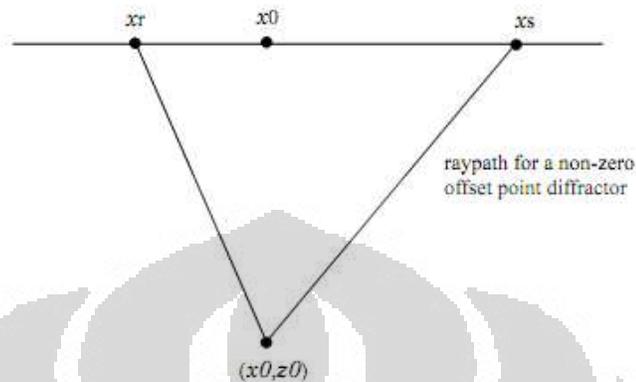
$$t_0 = \frac{2z_0}{v} \quad (2.6)$$

Pada kasus *non zero offset*, kurva difraksi didefinisikan sebagai persamaan yang dikenal sebagai persamaan *double square root*

$$t = \sqrt{t_0^2 + \frac{(x-x_r)^2}{v_{RMS}^2}} + \sqrt{t_0^2 + \frac{(x-x_s)^2}{v_{RMS}^2}} \quad (2.7)$$

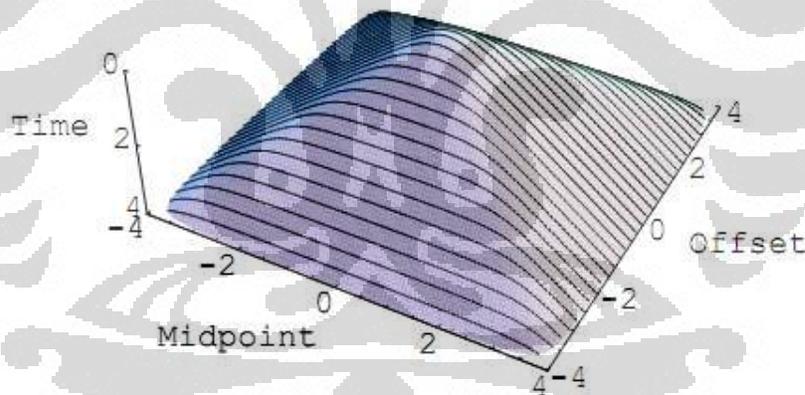
Dimana x_s merupakan posisi dari sumber dan x merupakan posisi penerima

Persamaan ini mendefinisikan waktu tempuh sepanjang *raypath* dari sumber x_s menuju titik difraksi (x_0, z_0) dan kembali menuju penerima x_r .



Gambar 2.12. *Raypath* untuk *non zero offset* (Paradigm, 1999)

Kurva permukaan difraksi *non zero offset* jika kita gambarakan akan menghasilkan yang kita kenal dengan *pyramid of cheops* (Clarbout, 1987).



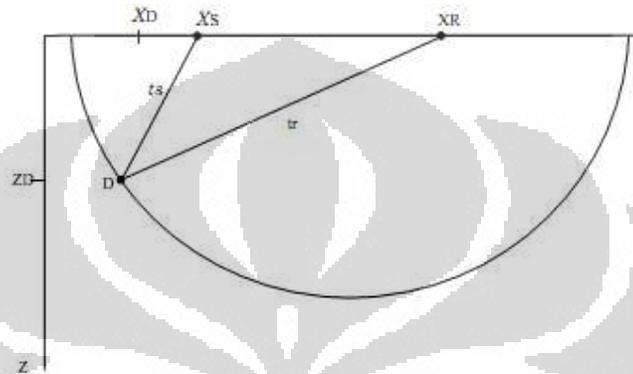
Gambar 2.13. Permukaan kurva difraksi untuk offset yang dibatasi. (Fomel, Sergey et al. 1999)

Migrasi *pre stack Kirchhoff* melibatkan penjumlahan *sample input* data sepanjang kurva difraksi *pre stack* dan meletakkan hasilnya pada puncak kurva (pada *zero offset*). Untuk migrasi waktu, kecepatan *RMS* dan persamaan *double square root* digunakan untuk menghitung permukaan kurva difraksi. Untuk migrasi kedalaman, *raypath* yang digunakan merupakan *raypath* sebenarnya hasil

ray tracing untuk setiap sumber dan setiap penerima digunakan untuk mendefinisikan permukaan difraksi.

2.7 Waktu Tempuh

Dengan migrasi *Kirchhoff*, setiap titik input data disebarkan menuju daerah *output* sepanjang permukaan respon impuls



Gambar 2.14. Data dari titik masukan yang disebarkan di *output space*.

(Fomel, Sergey et al. 1999)

Titik data pada waktu $t = t_r + t_s$ dari suatu tras dengan posisi sumber berada pada x_s dan penerima terletak pada x_r dipetakan oleh migrasi *pre stack*, ke titik D di sepanjang ellips. Satu hal yang sama untuk setiap titik D adalah waktu tempuh dari x_s menuju x_r yang melalui D sebanding dengan waktu *input* t . Untuk migrasi waktu, waktu tempuh t_r dan t_s dihitung menggunakan *straight rays* dan kecepatan *RMS*. Migrasi kedalaman sangat memperhitungkan kompleksitas dari model kecepatan dan menggunakan *raypath* yang sebenarnya dari x_s menuju D dan dari x_r menuju D. Migrasi kedalaman *pre stack Kirchhoff* menggunakan waktu tempuh sebenarnya dari setiap sumber dan penerima di permukaan untuk setiap titik dibawah permukaan yang terletak dalam batasan gambaran migrasi

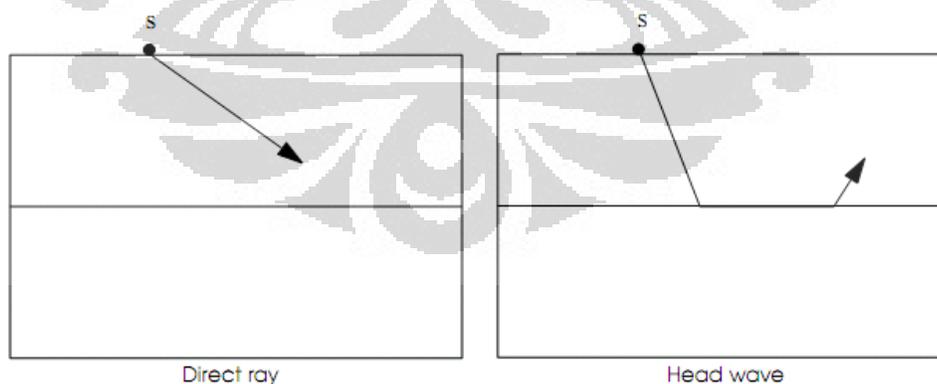
Migrasi kedalaman *pre stack Kirchhoff* merupakan program yang sangat kompleks, dikarenakan kebutuhannya untuk menghitung waktu tempuh. Menyimpan tabel waktu tempuh, yang merupakan data base yang sangat besar, serta menambahkan factor kompleksitas untuk migrasi *pre stack*

Ada beberapa cara untuk menghitung waktu tempuh. Migrasi kedalam *pre stack Kirchhoff* merupakan algoritma yang paling umum. Dua metode yang paling umum digunakan adalah:

- *Ray Tracing*
- Persamaan *eikonal*

Keduanya merupakan metode yang umum, sehingga akan bervariasi dalam pengaplikasian algoritmanya. *Ray tracing* merupakan prosedur yang dilakukan untuk melacak *raypath* sebenarnya antara 2 titik melalui model kecepatan yang kompleks. Waktu tempuh sepanjang *raypath* dihitung dengan mengintegrasikan fungsi kecepatan sepanjang sinar. Metode *eikonal* cukup berbeda. Metode ini tidak menggunakan fungsi *tracing raypath*, akan tetapi menghitung langsung dengan persamaan *eikonal*. Persamaan ini mendiskripsikan perubahan waktu tempuh sebagai fungsi lokasi untuk setiap nilai kecepatan dari suatu *medium*.

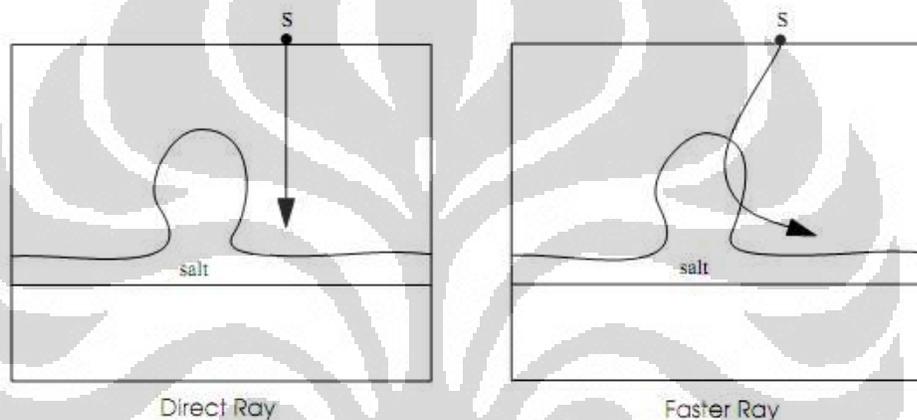
Variasi metode tidak selalu menghasilkan hasil yang sama. Pada banyak keadaan ada lebih dari satu sinar yang dapat berjalan antara dua titik. Perbedaan antara metode sering kali berarti memilih salah satu sinar yang lain. Masalah pemilihan *raypath* yang diinginkan untuk migrasi *Kirchhoff* merupakan masalah yang cukup kompleks, dan sangat bergantung kepada algoritma yang digunakan untuk perhitungan waktu tempuh. Beberapa program (program berbasis *eikonal*) memilih *raypath* yang tiba pertama, dimana biasanya merupakan gelombang langsung. Akan tetapi, terkadang *head wave* tiba terlebih dahulu.



Gambar 2.15. 2 tipe sinar yang mungkin akan muncul

Untuk alasan itu, kebanyakan program yang berbasiskan pendekatan *eikonal* memiliki masalah akurasi di bagian dangkal dan bagian yang terlebar dari batasan *aperture*, dimana *head waves* lebih cepat dari gelombang langsung. Pendekatan *ray tracing* (metode rekonstruksi muka gelombang) biasanya dapat mengatasi permasalahan ini, karena metode ini tidak bergantung kepada siapa yang tiba terlebih dahulu.

Situasi yang sama dapat diaplikasikan pada kubah garam, dimana dua jenis sinar kemungkinan akan muncul, satu yang berjalan secara langsung, dan yang lainnya yang berjalan melalui bagian garam yang lebih cepat.



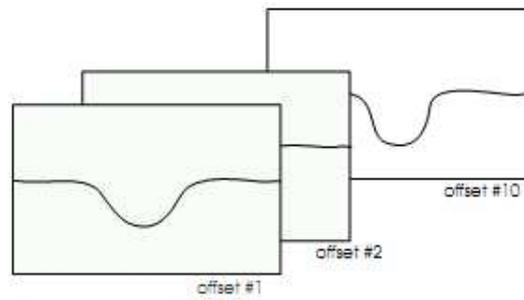
Gambar 2.16. 2 tipe sinar yang mungkin akan muncul pada kasus kubah garam (Paradigm, 1999)

Kembali, pemilihan algoritma yang berbeda akan menyebabkan pemilihan sinar yang berbeda. Metode rekonstruksi muka gelombang, memilih sinar yang membawa energi terbesar daripada sinar yang tercepat. Dengan melakukan *ray tracing* memungkinkan untuk memperhitungkan (dengan program migrasi *Kirchhoff*) banyak sinar daripada hanya memilih satu saja.

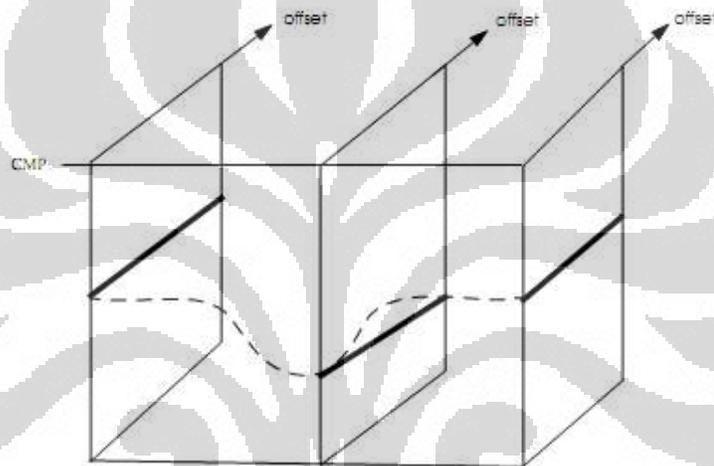
Dari contoh diatas kita dapat menarik kesimpulan bahwa pemilihan algoritma untuk perhitungan waktu tempuh sangat mungkin memiliki pengaruh yang signifikan pada hasil akhirnya. Sangat penting untuk memahami migrasi *Kirchhoff*, dengan kemampuannya menjumlahkan jumlah sinar yang cukup besar menjadi 1 titik. Maka dari itu, ada sebagian kecil sinar yang secara tidak langsung memiliki efek signifikan pada hasil akhir.

2.8 Migrasi *Pre stack Kirchhoff* common offset and analisis kecepatan migrasi.

Migrasi *pre stack Kirchhoff* dapat dilakukan dengan menjumlahkan semua data input point sepanjang kurva difraksi *pre stack*. Prosedur penjumlahan dapat dilakukan dalam satu tahapan saja. Tapi akan lebih menguntungkan untuk melakukan migrasi untuk setiap bidang *offset* secara terpisah, dan hanya dengan menjumlahkan semua *offset* untuk mendapatkan hasil migrasinya. Pemisahan seperti itu hanya dimungkinkan dengan migrasi *Kirchhoff*, karena bidang *common offset* tidak merepresentasikan *physical* eksperimen, dan juga karena migrasi *Kirchhoff* pada dasarnya hanya proses penjumlahan yang tidak kompleks. Dalam penjumlahan, susunan operasi tidak begitu penting. Maka dari itu migrasi *Kirchhoff pre stack* biasanya dilakukan dalam dua tahapan. Pertama melibatkan penjumlahan titik data yang memiliki *offset* yang sama dan langkah berikutnya adalah melibatkan penjumlahan antara *offset* (*stacked*). Langkah ini menguntungkan karena dapat digunakan untuk melakukan analisis kecepatan.



a) Individual migrated offset planes

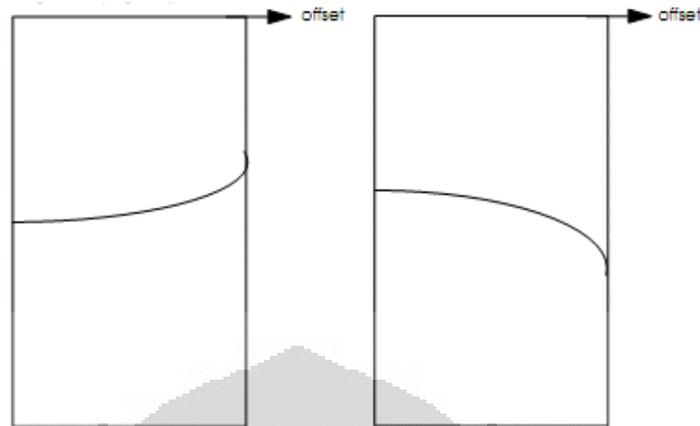


b) flat image gathers after pre-stack migration

Gambar 2.17. Two steps migrasi Kirchhoff. (Paradigm, 1999)

Gambar 2.17 mengilustrasikan dua langkah proses migrasi. Saat model kecepatan yang digunakan benar, setiap bidang *offset* akan termigrasi dengan benar. Hasil *gather* akan terlihat lurus pada eventnya sehingga dapat dilakukan proses *stack*.

Akan tetapi jika kecepatan yang digunakan tidak tepat, *CRP gather* tidak akan tersusun dengan tepat. Saat kecepatan terlalu lambat, waktu tempuh hasil perhitungan sepanjang sinar akan terlalu lama, efek ini akan sangat terasa pada data-data di *far offset*. Maka dari itu *event* yang terdapat pada *CRP* akan terdorong ke atas. Dan sebaliknya jika kecepatan yang digunakan terlalu cepat maka akan menghasilkan *undercorrected gather*.



Gambar 2.18. Pengaplikasian kecepatan migrasi yang kurang tepat.

Migrasi *Kirchhoff* menjadi migrasi yang umum digunakan untuk migrasi *pre stack* karena

- Merupakan algoritma *pre stack* migrasi tercepat
- Dapat melakukan migrasi pada dimensi *common offset*
- Dapat digunakan pada data yang tidak regular, karena migrasi dilakukan *trace by trace*. Ini merupakan keunggulan migrasi *pre stack* karena pada umumnya data *pre stack* tidak regular.

2.9 Definisi Migrasi CRAM

Common Reflection Angle Migration (CRAM) merupakan migrasi *multi-arrival*, migrasi yang bekerja berdasarkan *ray tracing*, menggunakan seluruh bagian gelombang dalam *aperture* yang terkontrol. Tidak seperti metode konvensional *ray tracing*, pada kasus ini *ray tracing* dilakukan dari titik *imaging* (dari segala arah, termasuk *turning rays*) naik menuju permukaan, membentuk suatu sistem yang berfungsi memetakan rekaman data seismik dipermukaan menjadi sesuatu yang dikenal sebagai *Local Angle Domain (LAD)* pada titik *imaging* (Koren et al, 2007). Prosedur ini bekerja berdasarkan iluminasi yang *uniform* dari segala arah pada titik *image*, memastikan bahwa semua sinar datang akan menjadi bahan perhitungan sementara keaslian amplitudo dan fase terjaga.

CRAM secara khusus didesain untuk melakukan seismik *imaging* dan tugas analisis dalam jumlah banyak, menentukan model kecepatan yang mendetail, orientasi pada target, *imaging* reservoir dengan resolusi tinggi, *AVA*

yang akurat dan ekstraksi properti reservoir, serta *imaging* data yang direkam didaerah dengan struktur dan kecepatan yang kompleks

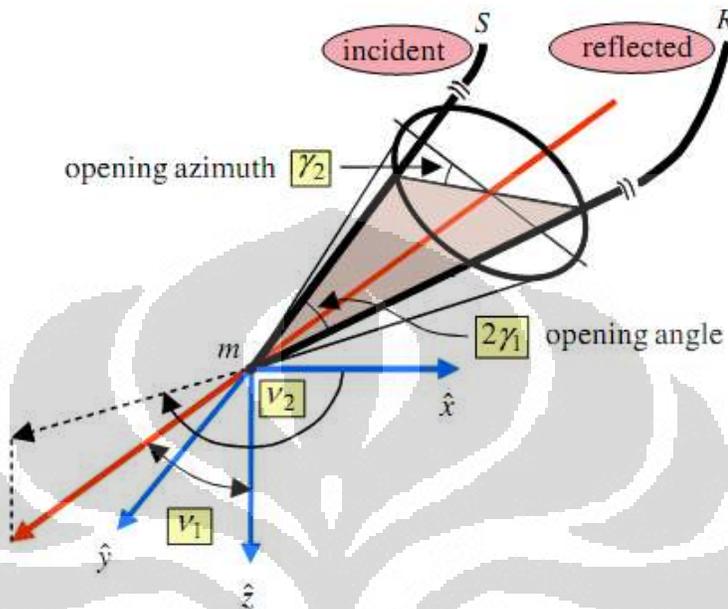
Migrasi *CRAM* seperti yang sudah disebutkan diatas memiliki banyak manfaat. *CRAM* bisa dilakukan pada data *full volume* dan *full aperture*, akan tetapi proses ini membutuhkan *cluster* dengan jumlah *node* yang sangat banyak. Juga bisa dilakukan secara spesifik hanya pada area tertentu yang menarik perhatian dibantu dengan latar belakang informasi kemiringan dan *azimuth*, dengan bantuan *aperture* yang sudah dimodelkan, membuat penyelesaiannya relatif lebih cepat dan performa kualitas tinggi dengan resolusi yang sangat baik

2.9.1. Metode CRAM

CRAM bekerja berdasarkan pendekatan *Local Angle Domain* (LAD) pada model bawah permukaan yang medianya dapat bersifat *isotropy* maupun *anisotropy*. Migrasi *asymptotic* atau operator *inverse*, termasuk didalamnya *ray path*, waktu tempuh, penyebaran geometri dan faktor rotasi fase (KMAH indek). Kesemuanya dihitung mulai dari titik *image* naik menuju ke permukaan .

Sistem *imaging* melibatkan interaksi dari 2 *wavefields* pada titik image, yaitu komponen normal *incident* dan refleksi/difraksi. Tiap *wavefield* dapat didekomposisikan menjadi gelombang bidang lokal lebih tepatnya digambarkan sebagai sinar, yang mengindikasikan arah penjalaran. Arah dari *incident* dan penyebaran sinar (*scattered ray*) secara konvensional mendeskripsikan sudut polaritasnya. Setiap sudut polar memiliki 2 komponen, yaitu kemiringan dan *azimuth*. Sehingga dibutuhkan 4 skalar sudut untuk mendefinisikan sistem dimensi sudut pada setiap titik *image*. Migrasi dimensi sudut yang menggunakan persamaan gelombang dan *ray tracing*, akan berhadapan dengan sistem sudut dimana keduanya, *incident* dan refleksi *wavefield* pada setiap titik *image* tersusun dari variasi arah dengan batasan yang lebar. Tahapan *imaging* melibatkan pengkombinasian jumlah pasangan sinar yang sangat besar (pasangan gelombang bidang lokal), merepresentasikan *incident* dan refleksi/difraksi. Setiap pasangan sinar memetakan rekaman data seismik dipermukaan yang bebas, menjadi dimensi 4D *Local Angle Domain* (Koren et al,2007). Dituliskan sebagai kemiringan v_1 dan azimuth v_2 dari pasangan normal *incident*, setengan sudut bukan γ_1 dan

azimuth bukaan γ_2 . Kita dapat menurunkan hubungan antara arah pasangan sinar dengan sudut *Local Angle Domain*.



Gambar 2.19. Contoh pasangan sinar (*incident* dan *scattered*) pada titik M dibawah permukaan dan 4 sudut yang berasosiasi dengan *Local Angle Domain*. (Koren et al, 2007)

Gambar 2.19 menunjukkan contoh hubungan dari pasangan sinar (*incident* dan *scattered*) dan 4 sudut yang berasosiasi dengan LAD. Sistem LAD memiliki kemungkinan untuk berasosiasi dengan sumbu *cross section* dari permukaan kerucut. Sumbu dari kerucut dengan pasangan sinar normal *incident* memiliki orientasi yang sama (kemiringan dan *azimuth*). Bukaan dari sudut sama dengan bukaan sudut dan orientasi dari *cross section* berkorespondensi dengan *azimuth* bukaan

2.9.2. Formula Migrasi

Ada 2 tipe *gather* domain sudut (*Angle Domain Common Image Gather*), berarah (*directional*) dan refleksi. Pada subsistem *directional*, reflectiviti pada titik *image* berfungsi sebagai pasangan sinar normal *zenith* v_1 dan *azimuth* v_2

$$I_v(m, v_1, v_2) = \frac{\cos^2 v_1}{v^2(m)} \int \frac{D_3(S, R, \tau_D)}{A(m, S)A(m, R)} \sin \gamma_1 dy_1 dy_2 \quad (2.8)$$

Pada subsistem refleksi reflectiviti I pada titik *image* berfungsi sebagai pasangan sinar normal sudut bukaan γ_1 dan bukaan azimuth γ_2

$$I_\gamma(m, \gamma_1, \gamma_2) = \frac{\cos^2 \gamma_1}{v^2(m)} \int \frac{D_3(S, R, \tau_D)}{A(m, S)A(m, R)} \sin v_1 dv_1 dv_2 \quad (2.9)$$

Dimana

$$S = S(m, v_1, v_2, \gamma_1, \gamma_2), \text{ dan } R = R(m, v_1, v_2, \gamma_1, \gamma_2), \quad (2.10)$$

Merupakan lokasi sumber dan penerima dipermukaan. Koordinat ini didapatkan melalui *ray tracing* yang dimulai pada titik *image*, dengan parameter sinar awal (kecepatan fase) yang berhubungan dengan sudut LAD (pendekatan *output-driven*). Parameter :

$$A(m, S) = \sqrt{\frac{V(m)}{8\pi|J(m, S)|}}, \quad A(m, R) = \sqrt{\frac{V(m)}{8\pi|J(m, R)|}} \quad (2.11)$$

Merupakan fungsi Green

$$D_3[S, R, \tau_D(m, S, R)] = \int iwU(S, R, w)^{i\Phi_3} dw \quad (2.12)$$

Merupakan data hasil proses filter

$$\Phi_3 = -w\tau_D(m, S, R) + \frac{\pi}{2}K(m, S, R)\text{sgn}(w) \quad (2.13)$$

Adalah fasenya, $U(S, R, w)$ merupakan tras seismik masukan, $K(m, S, R)$ merupakan KMAH index, $\tau_D(m, S, R)$ merupakan stack difraksi dimensi waktu. $V(m)$ merupakan fungsi dari parameter medium. Dalam kasus ini model isotropic bersesuaian dengan model kecepatan. J , merupakan penyebaran geometri. Persamaan 2.8 dan 2.9 mendeskripsikan pendekatan *output driven*, dimana data masukan seismik akan dimigrasikan menjadi fungsi sudut *LAD* pada titik *image*. Sehingga, secara teori, setiap pasangan sinar digunakan dalam migrasi membutuhkan kebebasan untuk menggunakan tras seismik manapun. Akses yang tidak beraturan pada data input membuat implementasinya prosesnya sangat sulit. Sebagai tambahan, pendekatan ini membutuhkan jumlah memori yang sangat besar untuk menyimpan data masukan

2.9.3 KMAH Index

Kebanyakan aplikasi yang menggunakan persamaan *dynamic ray tracing* membutuhkan solusi $q(\sigma)$ dan $p(\sigma)$ dari persamaan

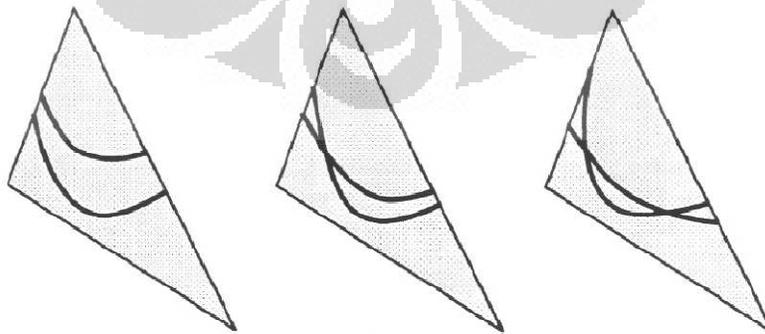
$$\frac{dq}{d\sigma} = p$$

$$\frac{dp}{d\sigma} = -\frac{3(S, xP_z - S, xP_x)^2}{4S^2} \quad (2.14)$$

Untuk keadaan awal dimana $q(\sigma)=0$ dan $p(\sigma)=1$. Menurut Cerveny et al, (1982) solusi yang berhubungan dengan keadaan awal ini biasanya diberi nama $q_2(\sigma)$ dan $p_2(\sigma)$. Dibutuhkan perhitungan *Gaussian beam*, sebagai tambahan, solusi yang berhubungan dengan $q(\sigma)=1$ dan $p(\sigma)=0$, biasanya dinamakan $q_1(\sigma)$ dan $p_1(\sigma)$. Solusi $q_1(\sigma)$ dan $p_1(\sigma)$ berhubungan dengan sumber gelombang bidang dan solusi untuk $q_2(\sigma)$ dan $p_2(\sigma)$ berhubungan dengan titik sumber.

Evaluasi dari fase sintetik seismogram membutuhkan adanya perhitungan jumlah nilai nol yang memotong fungsi $q_2(\sigma)$. Jumlah ini lah yang disebut *KMAH index* (e.g., Chapman, 1985) dan sebanding dengan jumlah sinar *paraxial* dari titik sumber yang berpotongan dengan *central ray*. Pergeseran fase $\frac{\pi}{2}$ terjadi dan berakumulasi setiap terjadi perpotongan.

Karena *raypath* berupa *parabolic* saat *sloth* bervariasi secara *linear*, kemungkinan untuk berpotongan 1x, 2x atau pun tidak sama sekali terjadi didalam suatu segitiga, seperti gambar dibawah.



Gambar 2.20. Kemungkinan yang dapat terjadi saat sinar memasuki segitiga. (Hale, Dave, 1991)

Menganggap q_0 dan p_0 menotasikan nilai dari $q_2(\sigma)$ dan $p_2(\sigma)$ saat suatu sinar memasuki segi tiga, membiarkan q dan p menotasikan nilai yang merepresentasikan saat sinar keluar dari segitiga, formula dibawah ini dapat digunakan untuk merevisi *KMAH index*:

$$\text{if } (q_0 q \geq 0 \text{ and } p_0 p < 0 \text{ and } q_0 p_0 < 0) \quad (2.15)$$

$$KMAH = KMAH + 2$$

$$\text{else if } (q = 0 \text{ or } q_0 q < 0 \text{ or } (q_0 = 0 \text{ and } p_0 p < 0 \text{ and } qp > 0)) \quad (2.16)$$

$$KMAH = KMAH + 1$$

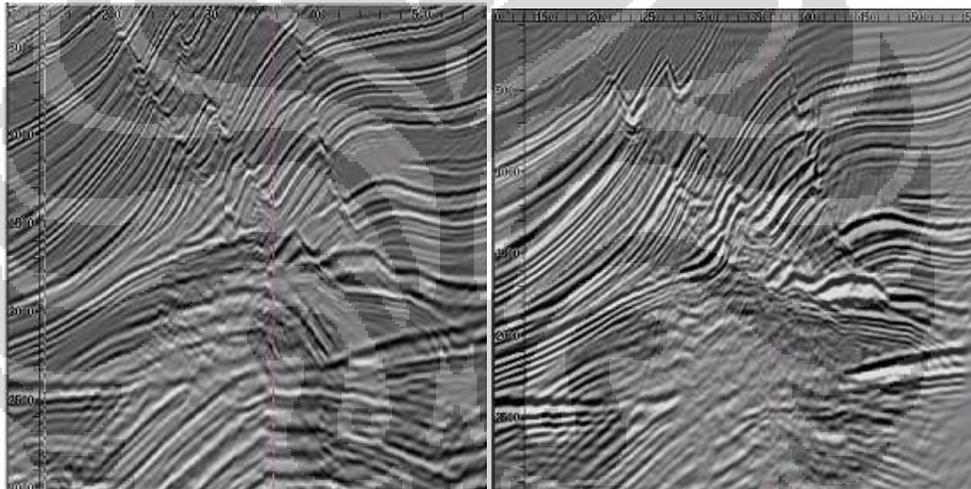
Persamaan ini didapatkan setelah dilakukan pencacahan semua kemungkinan yang dapat terjadi saat sinar berpotongan dalam suatu segitiga, termasuk ilustrasi pada gambar diatas, juga memperhitungkan kemungkinan tidak perpotongan yang terjadi tepat pada titik disaat akan masuk atau keluar dari segitiga. Bagi mereka yang ingin memverifikasi persamaan diatas perlu diperhatikan bahwa sinar *divergen* saat $pq > 0$, *konvergen* saat $pq < 0$ dan *parallel* saat $p = 0$

BAB 3

APLIKASI ALGORITMA LOCAL ANGLE DOMAIN

3.1. Aplikasi Pada Data Sintetik

Data sintetik yang digunakan adalah data set *Marmousi*. Data ini merupakan data sintetik yang umum digunakan untuk melakukan testing algoritma migrasi yang baru. Data ini merupakan data sintetik 2D yang mengacu pada geologi di daerah Cuanza basin, Angola. Strukturnya didominasi oleh growth fault diatas kubah garam, sehingga menghasilkan daerah dengan variasi kecepatan yang sangat kompleks. Target utama dari imaging pada daerah ini adalah daerah reservoir pada struktur antiklin dibawah kubah garam.

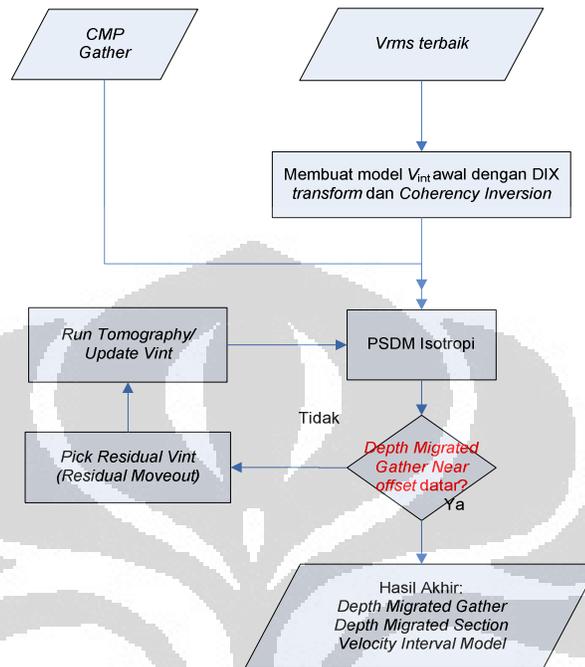


Gambar 3.1. Gambar *stack section* hasil migrasi dimensi kedalaman. (A). Hasil migrasi *CRAM*. (B). Hasil migrasi *Kirchhoff*

Gambar 3.1 menunjukkan hasil migrasi dengan menggunakan algoritma Local Angle Domain pada gambar A dan menggunakan Algoritma *Kirchhoff* pada gambar B.

Migrasi dapat langsung dilakukan tanpa melakukan tahapan persiapan disebabkan data sintetik yang digunakan sudah bersih dari *noise-noise* yang mengganggu, dan sudah dilengkapi dengan kecepatan *interval* yang akurat

3.2. Aplikasi Pada Data Real



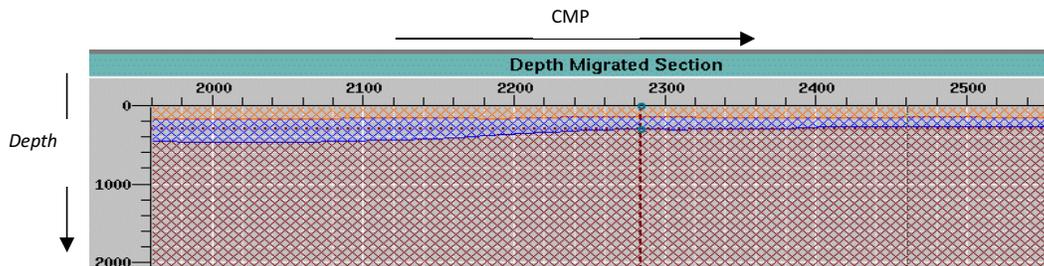
Gambar 3.2. Flowchart penelitian

Dalam pengolahan PSDM ada dua tahapan utama yaitu pemodelan dalam dimensi waktu dan pemodelan dalam dimensi kedalaman. Pemodelan dalam dimensi waktu dilakukan dengan menggunakan proses *PSTM* sampai didapatkan V_{rms} terbaik.

Transformasi kecepatan dari V_{rms} menjadi V_{int} dilakukan dengan 2 proses, yaitu:

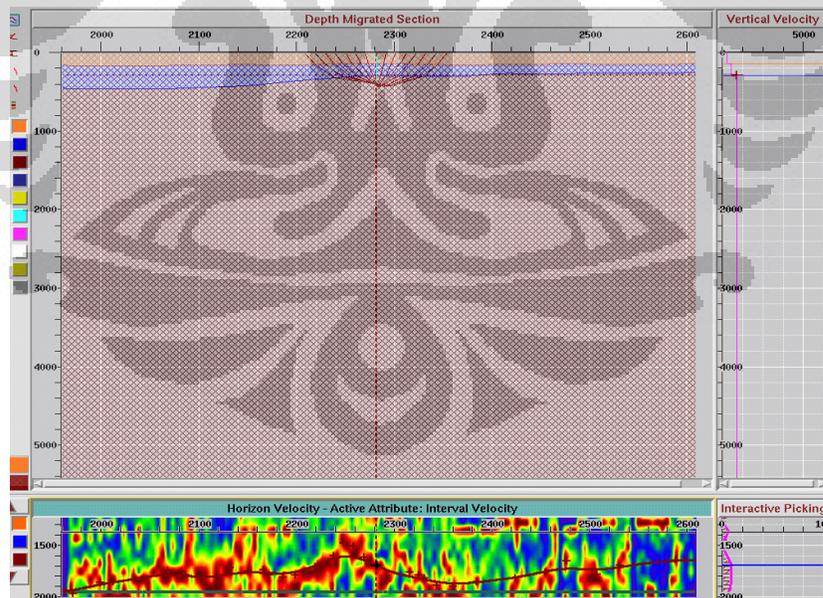
1. Transformasi Dix untuk 2 *horizon* pertama
2. *Coherency Inversion* untuk *horizon-horizon* berikutnya

Proses transformasi kecepatan ini menggunakan persamaan *Dix* (1955) untuk mendapatkan kecepatan interval. Persamaan ini hanya berlaku untuk kasus *horizon* yang *flat* atau lurus dan kecepatan lateral yang *homogen*, sehingga persamaan ini hanya diterapkan pada *horizon* yang terletak dekat dengan permukaan dalam kasus ini pada dua *horizon* teratas.



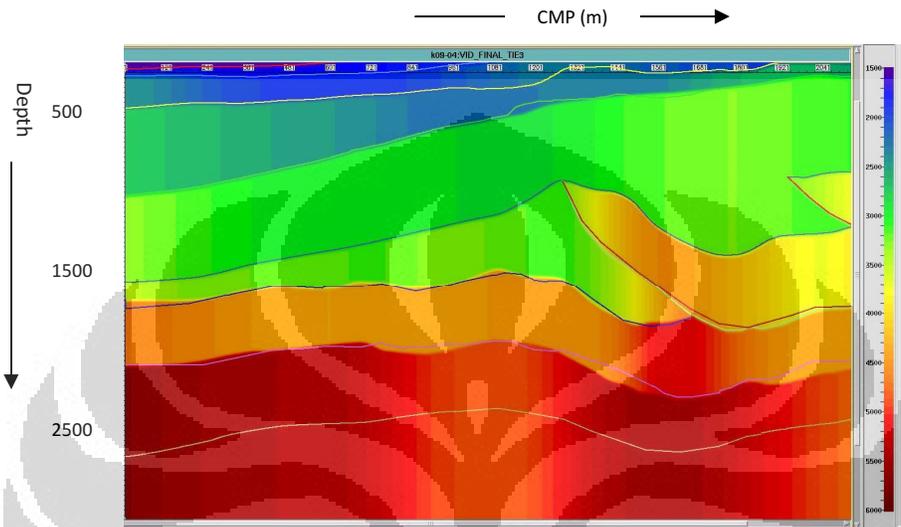
Gambar 3.3. Model poligon V_{interval} untuk dua lapisan teratas, menggunakan transformasi Dix

Selanjutnya adalah menggunakan *coherency inversion* untuk *horizon* ketiga sampai *horizon* terakhir. Cara kerja *coherency inversion* adalah hasil *picking horizon* pada domain waktu akan dijadikan referensi awal sebagai model struktur yang akan dikenai proses *ray tracing*. Hasil dari *ray tracing* model kecepatan pada kedalaman digunakan untuk mendapatkan *moveout* yang sesuai untuk *gather*. *Moveout* digambarkan sebagai *semblance* maksimum pada setiap CRP *gather*, yang akan mempresentasikan kecepatan interval terbaik. Pada saat *picking semblance* kontrol kualitas *picking* dapat dilihat pada *QC Time Gate window*. Saat *semblance* di-*picking* nilai kecepatan terbaik adalah nilai kecepatan yang mampu membuat *gather* lurus **gambar 3.4**.



Gambar 3.4. Proses *Coherency Inversion* untuk Horizon 3

Hasil akhir yang didapatkan dari proses *coherency inversion* ini merupakan model awal dari V_{int} . Kecepatan interval merupakan salah satu *input* untuk PSDM isotropi. Pada **gambar 3.5** dapat dilihat V_{int} yang merupakan hasil dari proses *Dix transform* dan *coherency inversion*.

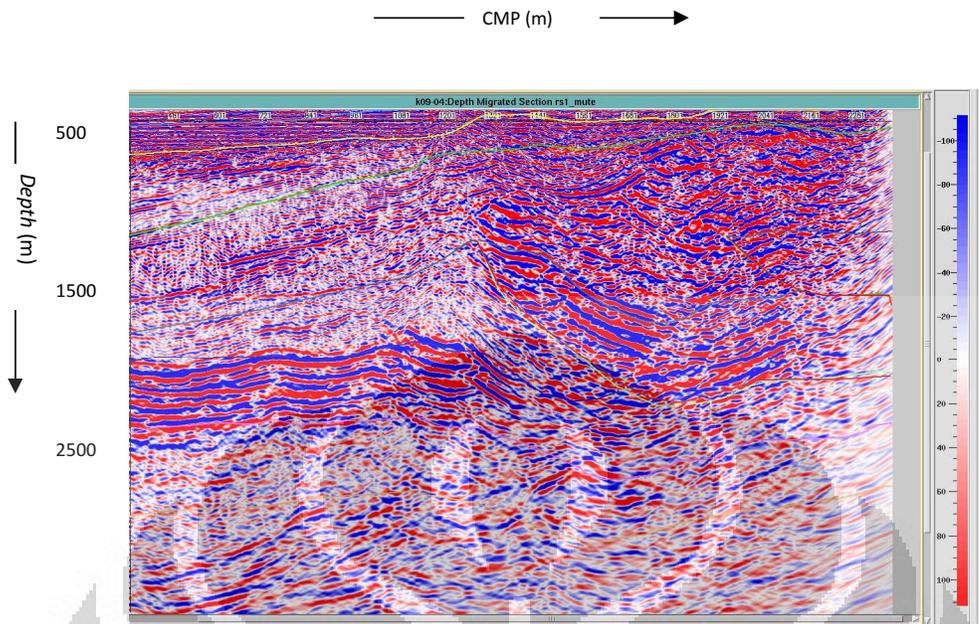


Gambar 3.5. V_{int} hasil transformasi Dix dan *Coherency Inversion*

3.3 *Pre Stack Depth Migration (PSDM)*

Migrasi dimensi kedalaman lebih unggul dibandingkan migrasi dimensi waktu karena migrasi dimensi kedalaman mampu mendeteksi variasi kecepatan lateral dan struktur geologi yang kompleks. Model V_{int} awal yang didapatkan dari proses transformasi *Dix* dan *coherency inversion*, serta *CMP Time Gather* merupakan masukan untuk proses PSDM.

Aperture merupakan parameter yang sangat penting dalam proses migrasi. *Aperture* harus dapat mencakup setiap reflektor yang menjadi target, agar amplitudo dapat dimigrasi keposisi reflektor sebenarnya. Lebar *aperture* merupakan jarak dari *far offset survey*. *Aperture* yang terlalu kecil akan membuat proses migrasi tidak dapat memberikan hasil yang maksimal karena tidak terjangkaunya reflektor target. Begitu juga sebaliknya, bila *aperture* yang terlalu besar, maka akan membuat proses migrasi berlangsung lebih lama. Hasil dari PSDM isotropi adalah *Depth Migrated Section* **gambar 3.6**.



Gambar 3.6. Penampang seismik kedalaman hasil PSDM

3.3.1 Memperbaiki Model Kecepatan dengan *Horizon Based Tomography*

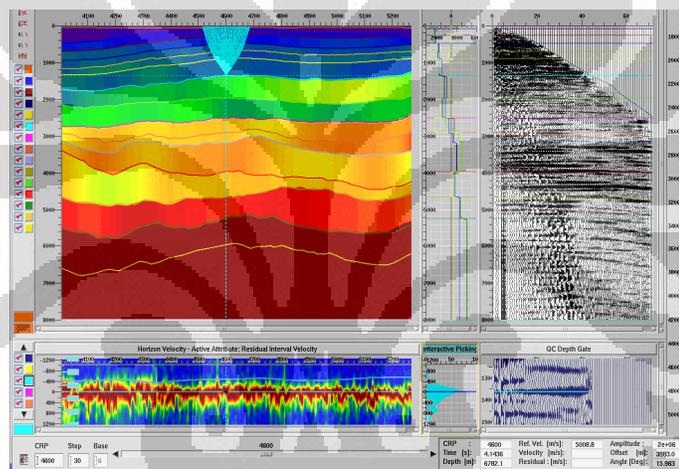
Perbaikan model struktur per lapisan yaitu dengan menggunakan fasilitas *structure model builder* pada perangkat lunak *Geodepth* dari data hasil *picking horizon* ulang penampang seismik dimensi kedalaman. Proses tersebut juga menghasilkan model kecepatan interval dalam dimensi kedalaman yang baru. Kecepatan interval merupakan parameter yang sangat penting dalam PSDM. Dengan konsep *global tomography*, maka kecepatan interval akan terus diperbaharui hingga diperoleh *depth gather* yang lurus atau *residual* V_{int} yang mendekati nol.

Horizon Based Tomography digunakan untuk memperkecil *errors* V_{int} . *Ray tracing* pada tiap lapisan digunakan untuk membuat *matrix tomography* sepanjang lintasan gelombang. *Error* dari tiap lapisan diselesaikan secara simultan menggunakan *least squares* untuk meminimalisasi kesalahan waktu tempuh yang melewati seluruh model. *Horizon Based Tomography* digunakan untuk menentukan kecepatan terbaik di masing-masing interval kecepatan.

Dengan melakukan *semblance plot* di sepanjang *horizon* yang telah di-*pick*, maka akan diperoleh residual dari kecepatan di setiap titik CRP di sepanjang

horizon. Jika *gather* dapat lurus atau *flat* maka *picking semblance* dianggap tepat (**gambar 3.7**). Kecepatan yang terlalu tinggi ditandai dengan *gather* pada *QC Depth Gate* melengkung ke bawah sedangkan kecepatan yang terlalu rendah ditandai dengan *gather* yang melengkung ke atas. Pada saat *picking semblance*, dipilih *semblance* dengan nilai maksimum sebagai nilai residual kecepatan interval.

Untuk mengetahui model kecepatan yang didapatkan telah benar, maka perlu melakukan pemeriksaan pada *gather* dan residual kecepatan. Model kecepatan sudah dianggap tepat jika *semblance* yang menunjukkan *residual* mendekati nol pada setiap titik CMP dan hasil dari *gather* PSDM isotropi atau *Depth Migrated gather* telah lurus



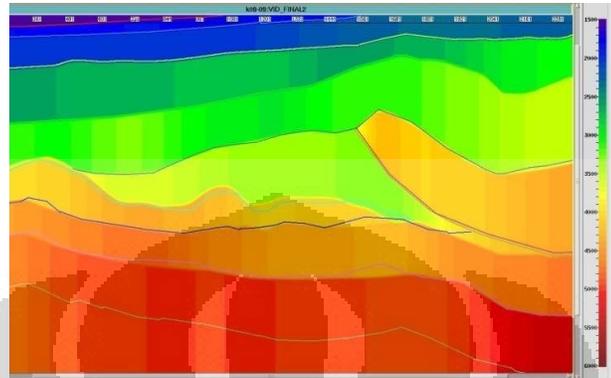
Gambar 3.7. Model based tomography pada horizon 6

3.4 Final PSDM

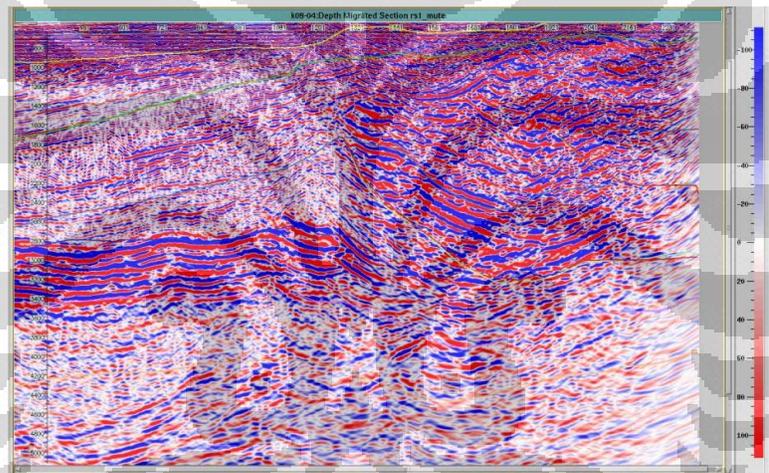
Proses PSDM dilakukan secara berulang kali, hal ini dilakukan untuk mengamati perubahan pencitraan (*imaging*). Dengan menggunakan V_{int} yang terus diperbaharui sampai didapatkan V_{int} terbaik (**gambar 3.8**), maka diharapkan diperoleh pencitraan seismik yang paling baik dan sesuai dengan struktur geologi sebenarnya.

Hasil dari PSDM *Depth Migrated Section* (**gambar 3.9**). Selain *depth migrated section*, PSDM menghasilkan *depth migrated gathers* (**gambar 3.10**). Migrasi CRAM yang menggunakan algoritma *Local Angle Domain* hanya digunakan pada proses paling akhir dengan anggapan bahwa kecepatan yang

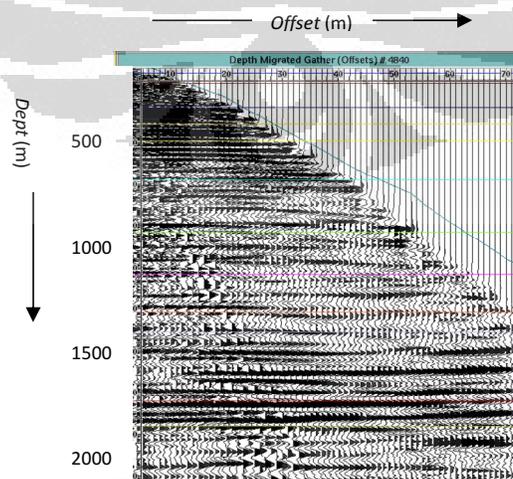
terakhir adalah yang paling tepat. Selain itu alasan yang paling utama adalah karena waktu komputasi yang lebih lama dibandingkan dengan migrasi menggunakan algoritma *Kirchhoff*



Gambar 3.8. V_{int} final hasil iterasi



Gambar 3.9. Stack section final hasil migrasi



Gambar 3.10. Depth Migrated Gather hasil dari PSDM

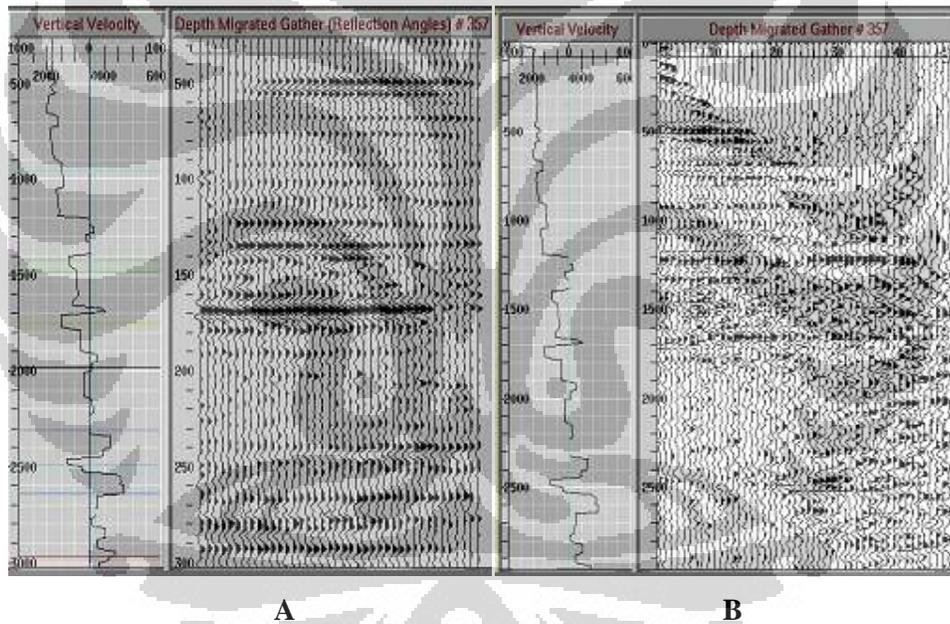
BAB 4

HASIL DAN ANALISIS

4.1 Analisis Data Sintetik

Pada bagian sebelumnya telah ditunjukkan gambar *section* hasil migrasi dimensi kedalaman dengan menggunakan migrasi *CRAM* dan *Kirchhoff*. Hasilnya menunjukkan keunggulan dari migrasi *CRAM* yang menggunakan algoritma *Local Angle Domain*. Untuk lebih jelasnya kita akan melihat contoh *CRP Gather* dari tiap masing-masing hasil migrasi.

Gambar 4.1 menunjukkan *CRP Gather* hasil migrasi *CRAM* pada gambar A dan *CRP gather* hasil migrasi *Kirchhoff* pada gambar B.

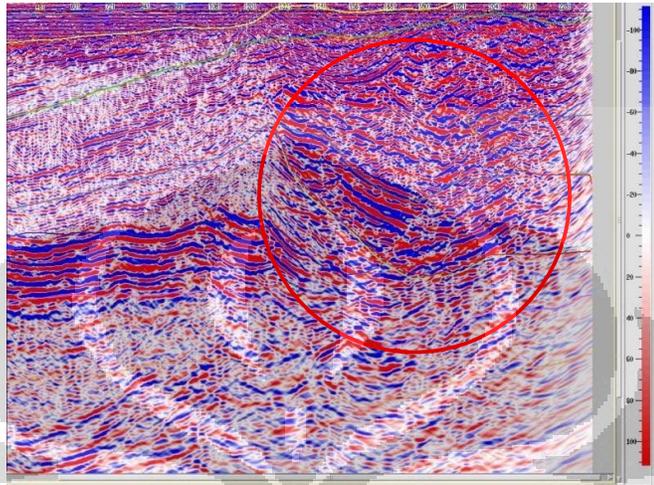


Gambar 4.1. *CRP Gather* hasil migrasi dimensi kedalaman. (A). Hasil migrasi *CRAM*. (B). Hasil migrasi *Kirchhoff*

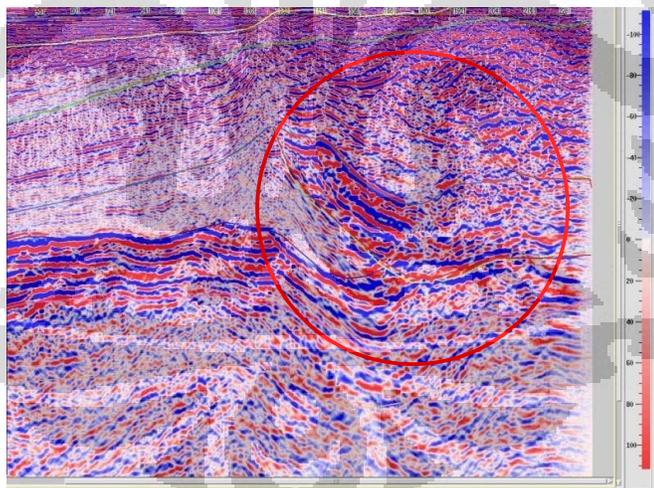
Dapat dilihat pada gambar A memiliki kualitas data (rasio S/N) yang lebih baik, hal ini disebabkan *spatial sampling* yang uniform pada dimensi sudut dibandingkan dengan *spatial sampling* pada dimensi offset yang pada umumnya irregular

4.2 Analisis Data Real

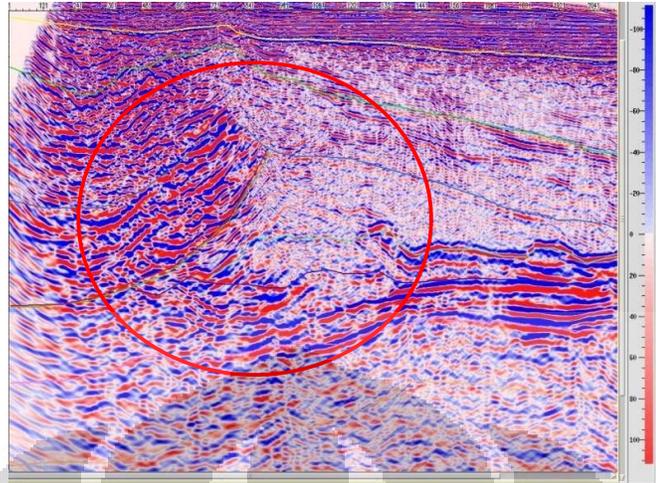
Dari hasil yang didapatkan terlihat bahwa secara umum migrasi dengan algoritma *LAD* lebih unggul dari pada migrasi dengan menggunakan algoritma *Kirchhoff*.



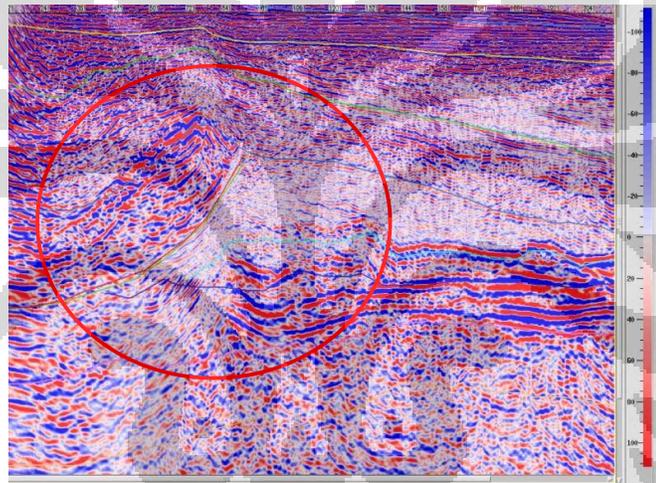
Gambar 4.2. Lintasan X1 Hasil PSDM dengan Algoritma *Kirchhoff*



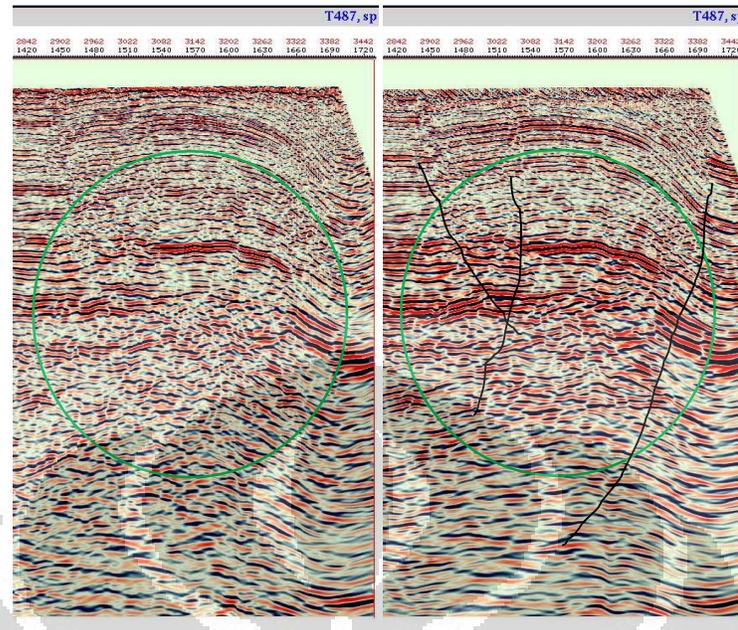
Gambar 4.3. Lintasan X1 Hasil PSDM dengan Algoritma *LAD*



Gambar 4.4. Lintasan X2 Hasil *PSDM* dengan Algoritma *Kirchhoff*



Gambar 4.5. Lintasan X2 Hasil *PSDM* dengan Algoritma *CRAM*



Gambar 4.6. Lintasan X3 Hasil PSDM dengan Algoritma *Kirchhoff* (gambar disebelah kiri) dan algoritma *CRAM* (gambar disebelah kanan)

Dari gambar - gambar diatas, pada **gambar 4.2** dan **gambar 4.3** terlihat perbedaan yang cukup signifikan yaitu pada daerah di dalam lingkaran, dimana hasil dari migrasi *CRAM* terlihat relatif lebih bersih dari *noise-noise* yang saling bertabrakan saat proses migrasi menyebabkan pada daerah tersebut menjadi lebih mudah untuk diinterpretasikan.

Sama seperti pada hasil sebelumnya, pada **gambar 4.4** dan **gambar 4.5**, hasil yang didapatkan juga menggambarkan keunggulan migrasi *CRAM* dibanding hasil migrasi *Kirchhoff*. Penyebab banyaknya *noise* (*swing/smile* akibat migrasi) adalah jika kita perhatikan kecepatan yang digunakan, terdapat variasi kecepatan *lateral* yang sangat berbeda sehingga migrasi dengan algoritma *Kirchhoff* kurang mampu menangani perubahan variasi kecepatannya

Juga pada **gambar 4.6** terlihat keunggulan dari migrasi *CRAM* pada area yang dilingkari, disana terlihat lebih mudah menginterpretasikan posisi-posisi patahan dan juga reflektor-reflektor yang terletak dibawah bidang patahan.

Dari hasil-hasil yang kita dapatkan dapat ditarik kesimpulan bahwa migrasi *CRAM* yang menggunakan algoritma *LAD* memiliki hasil yang lebih baik dibandingkan dengan menggunakan algoritma *Kirchhoff*. Yang menyebabkan

perbedaan ini adalah algoritma *Kirchhoff* hanya menggunakan *single raypath*, sementara *CRAM* menggunakan *multipath*. *Multipath* yang dimaksud adalah sinar-sinar yang perlu diikuti sertakan dalam perhitungan *travel time* dari titik *image* dibawah permukaan menuju penerima dipermukaan. Termasuk diantara sinar-sinar tersebut diantaranya:

- Yang berjalan langsung dari sumber menuju target reflektor kemudian kembali ke penerima.
- Yang terefleksikan dari reflektor
- Yang terefraksikan dari reflektor

Penyebab perlunya memasukkan ketiga sinar diatas dalam perhitungan adalah sinar refleksi yang langsung dan memiliki sudut tangensial akan saling berinterferensi yang bersifat destruktif disebabkan keduanya memiliki nilai amplitude yang relatif sebanding. Hal ini telah dibuktikan dalam suatu eksperimen dan perhitungan dengan menggunakan bidang planar yang terletak diantara medium dengan kecepatan rendah dan tinggi.

Alasan lainnya kenapa hasil migrasi *CRAM* lebih baik dari migrasi *Kirchhoff* adalah *imaging* dengan pendekatan *multipath* memiliki hasil yang optimal jika dilakukan pada dimensi *common angle reflection*. Alasan paling mendasar yang menjadi penyebabnya adalah untuk setiap titik *image*, kemiringan dan sudut refleksi, *raypath* dari sumber dan penerima menuju titik *image* didefinisikan secara unik. Hal ini tidak terjadi jika pendekatan *multipath* dilakukan pada dimensi *common offset*.

BAB 5

KESIMPULAN

Dengan metode *CRAM* dapat dihasilkan suatu sistem yang sangat baik dalam menghasilkan *image* pada dimensi kedalaman, walaupun *gather* yang dihasilkan dalam dimensi sudut. Sistem ini memiliki kemampuan untuk mengekstrak suatu informasi dengan resolusi yang baik mengenai model bawah permukaan. Baik pada objek yang memiliki struktur yang menerus hingga ke permukaan dan objek - objek dengan skala kecil, seperti rekahan, sehingga keberadaannya masih dapat terdeteksi, walaupun terletak dibawah struktur geologi yang kompleks. Berdasarkan eksperimen di atas didapatkan beberapa kesimpulan :

- Metode *CRAM* mampu mengatasi masalah imaging pada daerah yang memiliki struktur yang komplek
- Metode *CRAM* memiliki kemampuan memecahkan permasalahan imaging pada daerah yang memiliki variasi kecepatan kearah lateral
- Selain itu karena migrasi *CRAM* menghasilkan *gather* dengan dimensi sudut, maka hal ini membuka kemungkinan untuk dilakukannya *AVA* analisis
- *Gather* sudut refleksi dapat menyediakan informasi dari *attribute* geometri seperti, kemiringan, *azimuth* dan kemenerusan, jika diproses lebih lanjut.
- *Gather* sudut refleksi juga menyediakan informasi mengenai residual *moveout* dan variasi *amplitude* untuk seluruh *azimuth*, sehingga dapat digunakan sebagai indikasi secara langsung keberadaan isotropi *azimuthal* dan rekahan.
- Karena berorientasi pada target, hasilnya merupakan suatu *image* reservoir dengan resolusi tinggi dan informasi lainnya dalam batasan sumur.
- Waktu yang dibutuhkan pada proses migrasi *CRAM* jauh lebih lama dari metode konvensional (migrasi *Kirchhoff*)
- Untuk pengerjaan data dalam jumlah besar, seperti data *pre stack* 3D maka akan dibutuhkan komputer dalam jumlah besar

Daftar Acuan

- Audebert, F., P. Froidevaux, H. Rakotoarisoa, and J. Svay-Lucas, 2002, Presented at the 72nd Annual International Meeting, SEG.
- Biondi, B., 2007, *Residual moveout in anisotropic angle-domain common-image gathers*: Geophysics, 72, S93–S103.
- Biondi, B., and W. Symes, 2004, *Angle-domain common-image gathers for migration velocity analysis by wavefield continuation imaging*, Geophysics, 69, 1283–1298.
- Brandsberg-Dahl, S., M. V. de Hoop, and B. Ursin, 1999, *Velocity analysis in the common scattering-angle/azimuth domain*: 69th Annual International Meeting, SEG, Expanded Abstracts, 1715–1718.
- de Hoop, M. V., and N. Bleistein, 1997, *Generalized radon transform inversions for reflectivity in anisotropic elastic media: Inverse Problems*, 13, 669–690.
- Koren, Z., I. Ravve, A. Bartana, and D. Kosloff, 2007, *Local angle domain in seismic imaging*: Presented at the 69th Annual International Conference and Exhibition.
- Koren, Z., X. Sheng, and D. Kosloff, 2002, *Target-oriented common-reflection angle migration*: Presented at the 72nd Annual International Meeting, SEG, Expanded Abstracts.
- Miller, D., M. Oristaglio, and G. Beylkin, 1987, *A new slant on seismic imaging: Migration and integral geometry*: Geophysics, 52, 943–964.
- Paradigm, 1999. *2D Depth Migration manual book*.
- Rosseau, V., J. Svay-Lucas, L. Nicoletis, and H. Rakotoarisoa, 2000, *3D true-amplitude migration by regularization in angle domain*: Presented at the 62nd Annual International Conference and Exhibition, EAGE.
- Sepwww.stanford.edu/data/media/public/sep/jon/optical/Gip/expref.gif
- Ursin, B., 2004, *Parameter inversion and angle migration in anisotropic elastic media*: Geophysics, 69, 1125–1142.
- Wu, R. S., and L. Chen, 2006, *Directional illumination analysis using beamlet decomposition and propagation*: Geophysics, 71, S1147–S1159.
- Xu, S., H. Chauris, G. Lambaré, and M. Noble, 2001, *Common-angle migration: A strategy for imaging complex media*: Geophysics, 66, 1877–1894.