

BAB 2

KAJIAN PUSTAKA

2.1 Bioinformatika

2.1.1 Pengertian Bioinformatika

Bioinformatika adalah ilmu yang mempelajari penerapan teknik komputasional untuk mengelola dan menganalisis informasi biologis. Bidang ini mencakup penerapan metode-metode matematika, statistika, dan informatika untuk memecahkan masalah-masalah biologis, terutama dengan menggunakan sekuens DNA dan asam amino serta informasi yang berkaitan dengannya. Contoh topik utama bidang ini meliputi basis data untuk mengelola informasi biologis, penyejajaran sekuens (sequence alignment), prediksi struktur untuk meramalkan bentuk struktur protein maupun struktur sekunder RNA, analisis filogenetik, dan analisis ekspresi gen. Secara umum, Bioinformatika dapat digambarkan sebagai: segala bentuk penggunaan komputer dalam menangani informasi-informasi biologi. Dalam prakteknya, definisi yang digunakan oleh kebanyakan orang bersifat lebih terperinci. Bioinformatika menurut kebanyakan orang adalah satu sinonim dari komputasi biologi molekuler (penggunaan komputer dalam menandai karakterisasi dari komponen-komponen molekuler dari makhluk hidup).

Istilah bioinformatics mulai dikemukakan pada pertengahan era 1980-an untuk mengacu pada penerapan komputer dalam biologi. Namun demikian, penerapan bidang-bidang dalam bioinformatika (seperti pembuatan basis data dan pengembangan algoritma untuk analisis sekuens biologis) sudah dilakukan sejak tahun 1960-an.

Kemajuan teknik biologi molekular dalam mengungkap sekuens biologis dari protein (sejak awal 1950-an) dan asam nukleat (sejak 1960-an) mengawali perkembangan basis data dan teknik analisis sekuens biologis. Basis data sekuens protein mulai dikembangkan pada tahun 1960-an di Amerika Serikat, sementara basis data sekuens DNA dikembangkan pada akhir 1970-an di Amerika Serikat dan Jerman (pada European Molecular Biology Laboratory, Laboratorium Biologi Molekular Eropa). Penemuan teknik sekuensing DNA yang lebih cepat pada pertengahan 1970-an menjadi landasan terjadinya ledakan jumlah sekuens DNA yang berhasil diungkapkan pada 1980-an dan 1990-an, menjadi salah satu pembuka jalan bagi proyek-proyek pengungkapan genom, meningkatkan kebutuhan akan pengelolaan dan analisis sekuens, dan pada akhirnya menyebabkan lahirnya bioinformatika.

Perkembangan internet juga mendukung berkembangnya bioinformatika. Basis data bioinformatika yang terhubung melalui internet memudahkan ilmuwan mengumpulkan hasil sekuensing ke dalam basis data tersebut maupun memperoleh sekuens biologis sebagai bahan analisis. Selain itu, penyebaran program-program aplikasi bioinformatika melalui internet memudahkan ilmuwan mengakses program-program tersebut dan kemudian memudahkan pengembangannya.

2.1.2 Bioinformatika Klasik

Sebagian besar ahli Biologi mengistilahkan mereka sedang melakukan Bioinformatika ketika mereka sedang menggunakan komputer untuk menyimpan, melihat atau mengambil data, menganalisa atau memprediksi komposisi atau struktur dari biomolekul. Ketika kemampuan komputer menjadi semakin tinggi maka proses yang dilakukan dalam Bioinformatika dapat ditambah dengan melakukan simulasi. Yang

termasuk biomolekul diantaranya adalah materi genetik dari manusia --asam nukleat-- dan produk dari gen manusia, yaitu protein. Hal-hal di ataslah yang merupakan bahasan utama dari Bioinformatika "klasik", terutama berurusan dengan analisis sekuen (sequence analysis).

Definisi Bioinformatika menurut Fredj Tekaia dari Institut Pasteur [TEKAIA2004] adalah: "metode matematika, statistik dan komputasi yang bertujuan untuk menyelesaikan masalah-masalah biologi dengan menggunakan sekuen DNA dan asam amino dan informasi-informasi yang terkait dengannya." Dari sudut pandang Matematika, sebagian besar molekul biologi mempunyai sifat yang menarik, yaitu molekul-molekul tersebut adalah polymer, rantai-rantai yang tersusun rapi dari modul-modul molekul yang lebih sederhana, yang disebut monomer. Monomer dapat dianalogikan sebagai bagian dari bangunan, dimana meskipun bagian-bagian tersebut berbeda warna dan bentuk, namun semua memiliki ketebalan yang sama dan cara yang sama untuk dihubungkan antara yang satu dengan yang lain.

Monomer yang dapat dikombinasi dalam satu rantai ada dalam satu kelas umum yang sama, namun tiap jenis monomer dalam kelas tersebut mempunyai karakteristik masing-masing yang terdefinisi dengan baik. Beberapa molekul-molekul monomer dapat digabungkan bersama membentuk sebuah entitas yang berukuran lebih besar, yang disebut macromolecule. Macromolecule dapat mempunyai informasi isi tertentu yang menarik dan sifat-sifat kimia tertentu. Berdasarkan skema di atas, monomer-monomer tertentu dalam macromolecule dari DNA dapat diperlakukan secara komputasi sebagai huruf-huruf dari alfabet, yang diletakkan dalam sebuah aturan yang telah diprogram sebelumnya untuk membawa pesan atau melakukan kerja di dalam sel. Proses yang diterangkan di atas terjadi pada tingkat molekul di dalam sel. Salah satu cara untuk

mempelajari proses tersebut selain dengan mengamati dalam laboratorium biologi yang sangat khusus adalah dengan menggunakan Bioinformatika sesuai dengan definisi "klasik" yang telah disebutkan di atas.

2.1.3 Bioinformatika Baru

Salah satu pencapaian besar dalam metode bioinformatika adalah selesainya proyek pemetaan genom manusia (Human Genome Project). Selesainya proyek raksasa tersebut menyebabkan bentuk dan prioritas dari riset dan penerapan bioinformatika berubah. Secara umum dapat dikatakan bahwa proyek tersebut membawa perubahan besar pada sistem hidup kita, sehingga sering disebutkan --terutama oleh ahli biologi bahwa kita saat ini berada di masa pascagenom. Selesainya proyek pemetaan genom manusia ini membawa beberapa perubahan bagi Bioinformatika, diantaranya: Setelah memiliki beberapa genom yang utuh maka kita dapat mencari perbedaan dan persamaan di antara gen-gen dari spesies yang berbeda. Dari studi perbandingan antara gen-gen tersebut dapat ditarik kesimpulan tertentu mengenai spesies-spesies dan secara umum mengenai evolusi. Jenis cabang ilmu ini sering disebut sebagai perbandingan genom (comparative genomics).

Sekarang ada teknologi yang didisain untuk mengukur jumlah relatif dari kopi atau cetakan sebuah pesan genetik (level dari ekspresi genetik) pada beberapa tingkatan yang berbeda pada perkembangan atau penyakit atau pada jaringan yang berbeda. Teknologi tersebut contohnya seperti DNA microarrays akan semakin penting. Akibat yang lain, secara langsung adalah cara dalam skala besar untuk mengidentifikasi fungsi-fungsi dan keterkaitan dari gen (contohnya metode yeast twohybrid) akan semakin

tumbuh secara signifikan dan bersamanya akan mengikuti bioinformatika yang berkaitan langsung dengan kerja fungsi genom (functional genomics).

Akan ada perubahan besar dalam penekanan dari gen itu sendiri ke hasil-hasil dari gen. Yang pada akhirnya akan menuntun ke: usaha untuk mengkatalogkan semua aktivitas dan karakteristik interaksi antara semua hasil-hasil dari gen (pada manusia) yang disebut proteomics; usaha untuk mengkristalisasi dan memprediksikan struktur-struktur dari semua protein (pada manusia) yang disebut structural genomics. Apa yang disebut orang sebagai research informatics atau medical informatics, manajemen dari semua data eksperimen biomedik yang berkaitan dengan molekul atau pasien tertentu mulai dari spektroskop massal, hingga ke efek samping klinis akan berubah dari semula hanya merupakan kepentingan bagi mereka yang bekerja di perusahaan obat-obatan dan bagian TI Rumah Sakit akan menjadi jalur utama dari biologi molekuler dan biologi sel, dan berubah jalur dari komersial dan klinikal ke arah akademis. Dari uraian di atas terlihat bahwa bioinformatika sangat mempengaruhi kehidupan manusia, terutama untuk mencapai kehidupan yang lebih baik. Penggunaan komputer yang notabene merupakan salah satu keahlian utama dari orang yang bergerak dalam TI merupakan salah satu unsur utama dalam Bioinformatika, baik dalam Bioinformatika "klasik" maupun Bioinformatika "baru".

2.1.4 Cabang-cabang yang Terkait dengan Bioinformatika

Dari pengertian bioinformatika baik yang klasik maupun baru, terlihat banyak terdapat cabang-cabang disiplin ilmu yang terkait dengan bioinformatika terutama karena bioinformatika itu sendiri merupakan suatu bidang interdisipliner. Hal tersebut

menimbulkan banyak pilihan bagi orang yang ingin mendalami bioinformatika. Di bawah ini akan disebutkan beberapa bidang yang terkait dengan bioinformatika.

2.1.4.1 Biophysics

Biologi molekuler sendiri merupakan pengembangan yang lahir dari biophysics. Biophysics adalah sebuah bidang interdisipliner yang mengaplikasikan teknik-teknik dari ilmu Fisika untuk memahami struktur dan fungsi biologi (British Biophysical Society). Sesuai dengan definisi di atas, bidang ini merupakan suatu bidang yang luas. Namun secara langsung disiplin ilmu ini terkait dengan Bioinformatika karena penggunaan teknik-teknik dari ilmu Fisika untuk memahami struktur membutuhkan penggunaan TI.

2.1.4.2 Computational Biology

Computational biology merupakan bagian dari Bioinformatika (dalam arti yang paling luas) yang paling dekat dengan bidang Biologi umum klasik. Fokus dari computational biology adalah gerak evolusi, populasi, dan biologi teoritis daripada biomedis dalam molekul dan sel. Tidak dapat dielakkan bahwa Biologi Molekul cukup penting dalam computational biology, namun itu bukanlah inti dari disiplin ilmu ini. Pada penerapan computational biology, model-model statistika untuk fenomena biologi lebih disukai dipakai dibandingkan dengan model sebenarnya. Dalam beberapa hal cara tersebut cukup baik mengingat pada kasus tertentu eksperimen langsung pada fenomena biologi cukup sulit.

Tidak semua dari computational biology merupakan Bioinformatika, seperti contohnya Model Matematika bukan merupakan Bioinformatika, bahkan meskipun dikaitkan dengan masalah biologi.

2.1.4.3 Medical Informatics

Menurut Aamir Zakaria [ZAKARIA2004] Pengertian dari medical informatics adalah "sebuah disiplin ilmu yang baru yang didefinisikan sebagai pembelajaran, penemuan, dan implementasi dari struktur dan algoritma untuk meningkatkan komunikasi, pengertian dan manajemen informasi medis."

Medical informatics lebih memperhatikan struktur dan algoritma untuk pengolahan data medis, dibandingkan dengan data itu sendiri. Disiplin ilmu ini, untuk alasan praktis, kemungkinan besar berkaitan dengan data-data yang didapatkan pada level biologi yang lebih "rumit" yaitu informasi dari sistem-sistem superselular, tepat pada level populasi di mana sebagian besar dari Bioinformatika lebih memperhatikan informasi dari sistem dan struktur biomolekul dan selular.

2.1.4.4 Cheminformatics

Cheminformatics adalah kombinasi dari sintesis kimia, penyaringan biologis, dan pendekatan data-mining yang digunakan untuk penemuan dan pengembangan obat (Cambridge Healthtech Institute's Sixth Annual Cheminformatics conference). Pengertian disiplin ilmu yang disebutkan di atas lebih merupakan identifikasi dari salah satu aktivitas yang paling populer dibandingkan dengan berbagai bidang studi yang mungkin ada di bawah bidang ini.

Salah satu contoh penemuan obat yang paling sukses sepanjang sejarah adalah penisilin, dapat menggambarkan cara untuk menemukan dan mengembangkan obat-obatan hingga sekarang. Cara untuk menemukan dan mengembangkan obat adalah hasil dari kesempatan, observasi, dan banyak proses kimia yang intensif dan lambat. Sampai beberapa waktu yang lalu, disain obat dianggap harus selalu menggunakan kerja yang intensif, proses uji dan gagal (trial-error process). Kemungkinan penggunaan TI untuk merencanakan secara cerdas dan dengan mengotomatiskan proses-proses yang terkait dengan sintesis kimiawi dari komponen-komponen pengobatan merupakan suatu prospek yang sangat menarik bagi ahli kimia dan ahli biokimia. Penghargaan untuk menghasilkan obat yang dapat dipasarkan secara lebih cepat sangatlah besar, sehingga target inilah yang merupakan inti dari cheminformatics.

Ruang lingkup akademis dari cheminformatics ini sangat luas. Contoh bidang minatnya antara lain: Synthesis Planning, Reaction and Structure Retrieval, 3-D Structure Retrieval, Modelling, Computational Chemistry, Visualisation Tools and Utilities.

2.1.4.5 Mathematical Biology

Mathematical biology lebih mudah dibedakan dengan Bioinformatika daripada computational biology dengan Bioinformatika. Mathematical biology juga menangani masalah-masalah biologi, namun metode yang digunakan untuk menangani masalah tersebut tidak perlu secara numerik dan tidak perlu diimplementasikan dalam software maupun hardware. Bahkan metode yang dipakai tidak perlu "menyelesaikan" masalah apapun; dalam mathematical biology bisa dianggap beralasan untuk mempublikasikan

sebuah hasil yang hanya menyatakan bahwa suatu masalah biologi berada pada kelas umum tertentu.

Menurut Alex Kasman [KASMAN2004] Secara umum mathematical biology melingkupi semua ketertarikan teoritis yang tidak perlu merupakan sesuatu yang beralgoritma, dan tidak perlu dalam bentuk molekul, dan tidak perlu berguna dalam menganalisis data yang terkumpul.

2.1.4.6 Proteomics

Istilah proteomics pertama kali digunakan untuk menggambarkan himpunan dari protein-protein yang tersusun (encoded) oleh genom. Ilmu yang mempelajari proteome, yang disebut proteomics, pada saat ini tidak hanya memperhatikan semua protein di dalam sel yang diberikan, tetapi juga himpunan dari semua bentuk isoform dan modifikasi dari semua protein, interaksi diantaranya, deskripsi struktural dari protein-protein dan kompleks-kompleks orde tingkat tinggi dari protein, dan mengenai masalah tersebut hampir semua pasca genom.

Michael J. Dunn [DUNN2004], Pemimpin Redaksi dari Proteomics mendefinisikan kata "proteome" sebagai: "The PROTEin complement of the genOME" Dan mendefinisikan proteomics berkaitan dengan: "studi kuantitatif dan kualitatif dari ekspresi gen di level dari protein-protein fungsional itu sendiri". Yaitu: "sebuah antarmuka antara biokimia protein dengan biologi molekuler".

Mengkarakterisasi sebanyak puluhan ribu protein-protein yang dinyatakan dalam sebuah tipe sel yang diberikan pada waktu tertentu --apakah untuk mengukur berat molekul atau nilai-nilai isoelektrik protein-protein tersebut-- melibatkan tempat

penyimpanan dan perbandingan dari data yang memiliki jumlah yang sangat besar, tak terhindarkan lagi akan memerlukan Bioinformatika.

2.1.4.7 Pharmacogenomics

Pharmacogenomics adalah aplikasi dari pendekatan genomik dan teknologi pada identifikasi dari target-target obat. Contohnya meliputi menjangkau semua genom untuk penerima yang potensial dengan menggunakan cara Bioinformatika, atau dengan menyelidiki bentuk pola dari ekspresi gen di dalam baik patogen maupun induk selama terjadinya infeksi, atau maupun dengan memeriksa karakteristik pola-pola ekspresi yang ditemukan dalam tumor atau contoh dari pasien untuk kepentingan diagnosa (kemungkinan untuk mengejar target potensial terapi kanker).

Istilah pharmacogenomics digunakan lebih untuk urusan yang lebih "trivial" tetapi dapat diargumentasikan lebih berguna dari aplikasi pendekatan Bioinformatika pada pengkatalogan dan pemrosesan informasi yang berkaitan dengan ilmu Farmasi dan Genetika, untuk contohnya adalah pengumpulan informasi pasien dalam database.

2.2 Kecerdasan Buatan

Kecerdasan buatan (Artificial Intelligence) adalah bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan. Lebih dari itu, komputer diharapkan dapat digunakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia baik dalam bidang pendidikan, kesehatan, industri, dan kehidupan sehari-hari sehingga peran komputer dan

manusia akan saling melengkapi. Beberapa hal yang menjadi kekurangan manusia diharapkan dapat digantikan oleh komputer. Begitu juga dengan komputer yang tak akan berguna tanpa sentuhan manusia.

2.2.1 Sejarah Kecerdasan Buatan

Pada awal abad 17, René Descartes mengemukakan bahwa tubuh hewan bukanlah apa-apa melainkan hanya mesin-mesin yang rumit. Blaise Pascal menciptakan mesin penghitung digital mekanis pertama pada 1642. Pada 19, Charles Babbage dan Ada Lovelace bekerja pada mesin penghitung mekanis yang dapat diprogram.

Bertrand Russell dan Alfred North Whitehead menerbitkan *Principia Mathematica*, yang merombak logika formal. Warren McCulloch dan Walter Pitts menerbitkan "Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas " pada 1943 yang meletakkan pondasi untuk jaringan syaraf.

Tahun 1950-an adalah periode usaha aktif dalam AI. Program AI pertama yang bekerja ditulis pada 1951 untuk menjalankan mesin Ferranti Mark I di University of Manchester (UK): sebuah program permainan naskah yang ditulis oleh Christopher Strachey dan program permainan catur yang ditulis oleh Dietrich Prinz. John McCarthy membuat istilah "kecerdasan buatan " pada konferensi pertama yang disediakan untuk pokok persoalan ini, pada 1956. Dia juga menemukan bahasa pemrograman Lisp. Alan Turing memperkenalkan "Turing test" sebagai sebuah cara untuk mengoperasionalkan test perilaku cerdas. Joseph Weizenbaum membangun ELIZA, sebuah chatterbot yang menerapkan psikoterapi Rogerian.

Selama tahun 1960-an dan 1970-an, Joel Moses mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah di dalam program Maccsya,

program berbasis pengetahuan yang sukses pertama kali dalam bidang matematika. Marvin Minsky dan Seymour Papert menerbitkan *Perceptrons*, yang mendemonstrasikan batas jaringan syaraf sederhana dan Alain Colmerauer mengembangkan bahasa komputer Prolog. Ted Shortliffe mendemonstrasikan kekuatan sistem berbasis aturan untuk representasi pengetahuan dan inferensi dalam diagnosa dan terapi medis yang kadangkala disebut sebagai sistem pakar pertama. Hans Moravec mengembangkan kendaraan terkendali komputer pertama untuk mengatasi jalan berintang yang kusut secara mandiri.

Pada tahun 1980-an, jaringan syaraf digunakan secara meluas dengan algoritma perambatan balik, pertama kali diterangkan oleh Paul John Werbos pada 1974. Tahun 1990-an ditandai perolehan besar dalam berbagai bidang AI dan demonstrasi berbagai macam aplikasi. Lebih khusus Deep Blue, sebuah komputer permainan catur, mengalahkan Garry Kasparov dalam sebuah pertandingan 6 game yang terkenal pada tahun 1997. DARPA menyatakan bahwa biaya yang disimpan melalui penerapan metode AI untuk unit penjadwalan dalam Perang Teluk pertama telah mengganti seluruh investasi dalam penelitian AI sejak tahun 1950 pada pemerintah AS.

Tantangan Hebat DARPA, yang dimulai pada 2004 dan berlanjut hingga hari ini, adalah sebuah pacuan untuk hadiah \$2 juta dimana kendaraan dikemudikan sendiri tanpa komunikasi dengan manusia, menggunakan GPS, komputer dan susunan sensor yang canggih, melintasi beberapa ratus mil daerah gurun yang menantang.

2.2.2 Beda Kecerdasan Buatan dan Kecerdasan Alami

- Kelebihan kecerdasan buatan :
 1. Lebih bersifat permanen. Kecerdasan alami bisa berubah karena sifat manusia pelupa. Kecerdasan buatan tidak berubah selama sistem komputer dan program tidak mengubahnya.
 2. Lebih mudah diduplikasi dan disebar. Menransfer pengetahuan manusia dari 1 orang ke orang lain membutuhkan proses yang sangat lama dan keahlian tidak akan pernah dapat diduplikasi dengan lengkap. Jadi jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.
 3. Lebih murah. Menyediakan layanan komputer akan lebih mudah dan murah dibandingkan mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
 4. Bersifat konsisten dan teliti karena kecerdasan buatan adalah bagian dari teknologi komputer sedangkan kecerdasan alami senantiasa berubah-ubah
 5. Dapat didokumentasi. Keputusan yang dibuat komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
 6. Dapat mengerjakan beberapa task lebih cepat dan lebih baik dibanding manusia

- Kelebihan kecerdasan alami :
 1. Kreatif : manusia memiliki kemampuan untuk menambah pengetahuan, sedangkan pada kecerdasan buatan untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.
 2. Memungkinkan orang untuk menggunakan pengalaman atau pembelajaran secara langsung. Sedangkan pada kecerdasan buatan harus mendapat masukan berupa input-input simbolik.
 3. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

2.2.3 Ruang Lingkup Kecerdasan Buatan

Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan peran kecerdasan buatan. Kecerdasan buatan tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah digunakan di beberapa disiplin ilmu lain. Misalnya pada bidang manajemen, adanya sistem pendukung keputusan dan sistem informasi manajemen juga tidak terlepas dari peran kecerdasan buatan. Adanya irisan penggunaan kecerdasan buatan di berbagai disiplin ilmu menyebabkan sulitnya pengklasifikasian kecerdasan buatan berdasarkan disiplin ilmu. Oleh karena itu, pengklasifikasian kecerdasan buatan dibuat berdasarkan keluaran yang dihasilkan. Ruang lingkup utama dalam kecerdasan buatan adalah:

- Sistem pakar

Sistem pakar dapat menyelesaikan permasalahan yang biasa diselesaikan oleh seorang pakar (Rich, 1991, p547). Komputer digunakan sebagai sarana untuk menyimpan basis

pengetahuan seorang pakar sehingga komputer dapat memiliki keahlian untuk menyelesaikan permasalahan layaknya seorang pakar.

- Pengolahan bahasa sehari-hari

Sebuah program yang mampu memahami bahasa manusia (Luger, 1993, p17). Komputer diberikan pengetahuan mengenai bahasa manusia sehingga pengguna dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.

- Perencanaan dan robotik

Perencanaan adalah sebuah aspek yang penting dalam merancang sebuah sistem robot yang dapat menyelesaikan tugas-tugasnya dengan beberapa derajat fleksibilitas dan tanggung jawab terhadap dunia luar (Luger, 1993, p19) .

- Visi komputer

Visi komputer dapat dideskripsikan sebagai sebuah deduksi otomatis dari struktur atau properti dari ruang dimensi tiga baik dari satu atau beberapa citra dua dimensi dari ruang lingkup dan pengenalan objek dengan bantuan properti ini. Tujuan dari visi komputer adalah untuk menarik kesimpulan mengenai lingkungan fisik dari citra yang ambigu atau yang memiliki derau (Kulkarni, 2001, p27).

- Permainan

Konsep kecerdasan buatan dapat diterapkan pada beberapa permainan seperti catur dan dam. Permainan-permainan ini memiliki aturan main yang jelas sehingga mudah untuk dapat mengaplikasikan teknik pencarian heuristik (Luger, 1993, p14). Pencarian heuristik adalah metode pencarian yang dilakukan dengan menggunakan penalaran. Dengan pencarian heuristik, mesin dapat memunculkan beberapa kemungkinan yang dapat dilakukan dan mencari jalan yang terbaik atau mendekati hasil yang diinginkan.

2.3 Visi Komputer

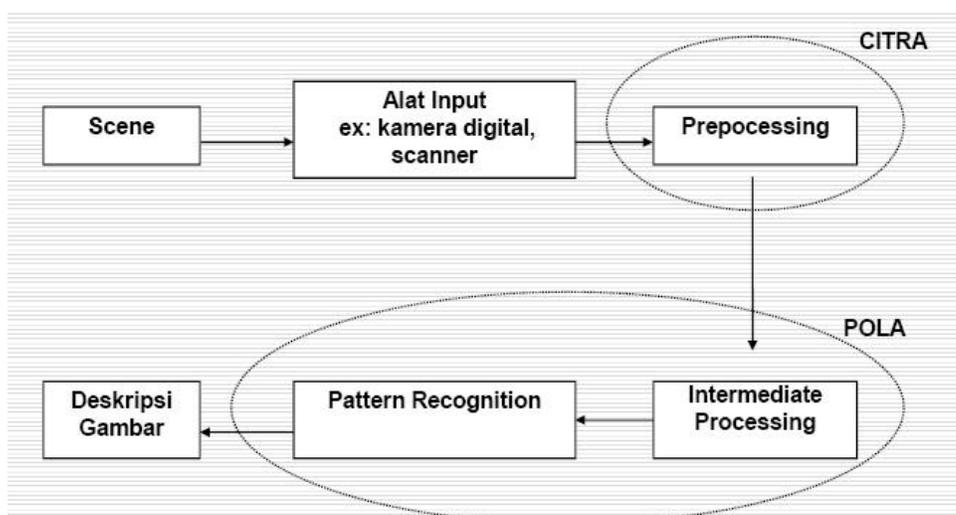
Computer visiin atau visi komputer sering didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati atau diobservasi. Cabang ilmu ini bersama Intelijensia Semu (Artificial Intelligence) akan mampu menghasilkan sistem intelijen visual (Visual Intelligence System).

Computer Vision adalah kombinasi antara Pengolahan Citra dan Pengenalan Pola.

Proses-proses dalam computer vision dapat dibagi menjadi 3 aktivitas :

1. Memperoleh atau mengakuisisi citra digital
2. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra (operasi-operasi pengolahan citra)
3. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya mengontrol peralatan, memantau proses manufaktur, memandu robot, dll.

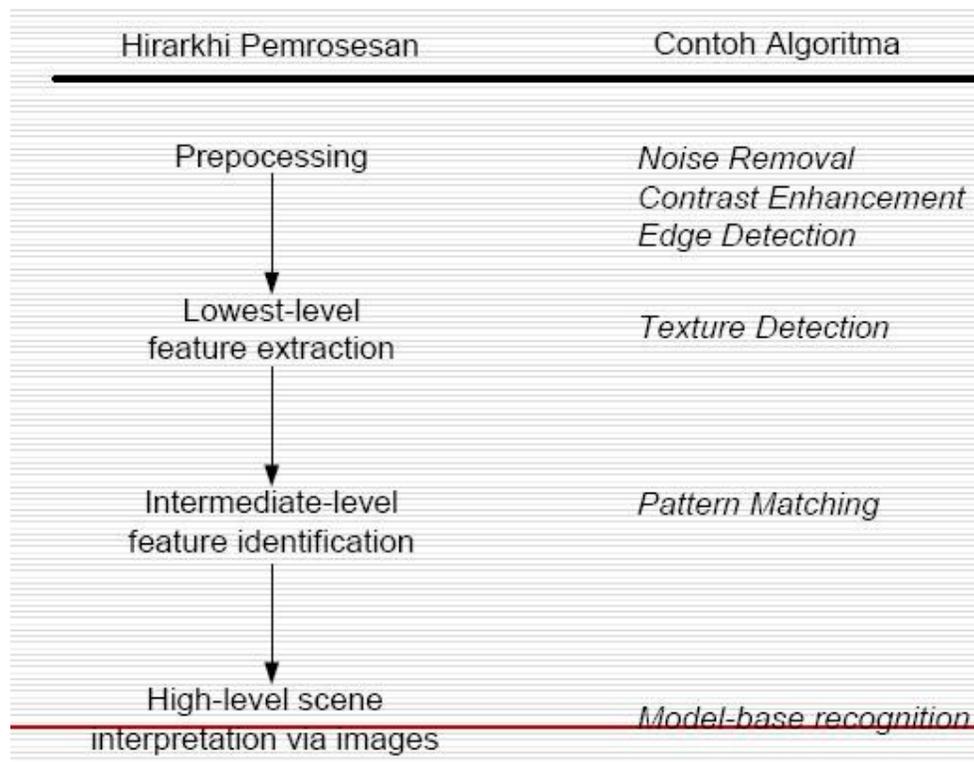
adapun skema computer vision dapat dllihat dari gambar berikut:



Gambar 2.1 Skema Computer Vision

dari gambar di atas dapat dilihat bagaimana sebuah skema kegiatan computer vision berawal dari scene atau image atau citra yang di ambil dengan bantuan alat input seperti kamera, scanner atau alat input lainnya, kemudian citra ini mengalami prapemrosesan dari citra tersebut di pindahkan ke komputer untuk menjadi sebuah pola dan mengalami intermediate processing. Image dimanipulasi kemudian di intepretasikan dalam pattern recognition sehingga mendapatkan hasil akhir sebuah deskripsi gambar, bisa berupa pengetahuan tentang bentuk matrik gambar tersebut yang nantinya bisa dimanipulasi lagi untuk keperluan pengontrolan atau pemantauan.

Untuk hierarki lebih jelasnya proses sebuah compter vision dapat dilihat dari gambar berikut :



Gambar 2.2 Hierarki Proses Computer Vision

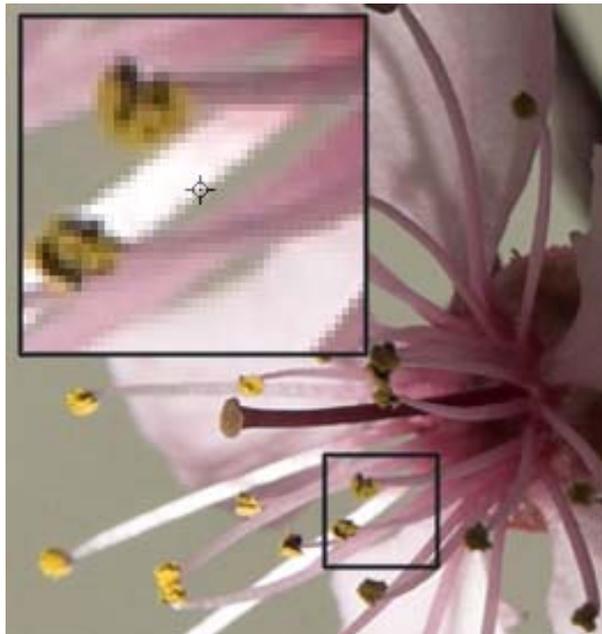
2.4 Citra

2.4.1 Pengertian Citra

Pada grafik design pengertian dari sebuah gambar atau citra dibedakan atas 2 jenis yaitu :

- Gambar Bitmap :

Gambar Bitmap sering disebut juga dengan gambar raster. Gambar Bitmap adalah gambar yang terbentuk dari pixel, dengan setiap pixelnya mempunyai warna tertentu. Jika gambar bitmap ini diperbesar, misalnya menjadi 4 kalinya, maka gambar akan menjadi kabur karena pixelnya juga bertambah besar menjadi 4 kalinya. (kualitas gambar menurun). Format gambar bitmap sering dipakai dalam foto dan gambar. Dua istilah yang perlu dipahami ketika bekerja dengan gambar bitmap adalah resolusi dan kedalaman warna. Gambar bitmap biasanya diperoleh dengan cara : Scanner, Camera Digital, Video Capture dll.



Gambar 2.3 Gambar bitmap

- Gambar Vektor :

Gambar Vektor dihasilkan dari perhitungan matematis dan tidak berdasarkan pixel. Jika gambar di perbesar atau diperkecil, kualitas gambar relatif tetap baik dan tidak berubah. Gambar vektor biasanya dibuat menggunakan aplikasi-aplikasi gambar vektor misalkan : Corel Draw, Adobe Illustrator, Macromedia Freehand, Autocad dll.

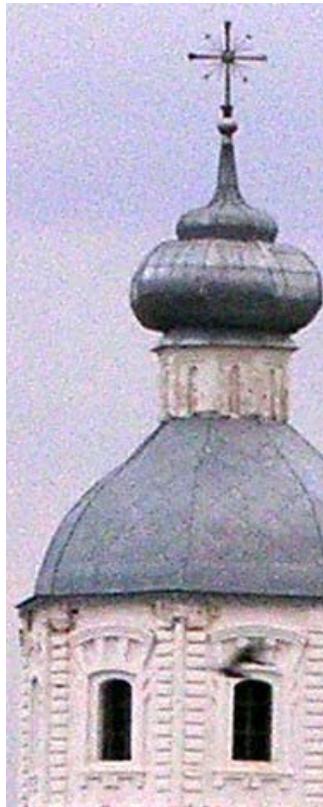


Gambar 2.4 Gambar vektor

2.4.2 Gangguan Citra

Image noise adalah setara digital film gabah untuk kamera analog. Kita dapat menganggapnya sebagai analog dengan latar belakang halus mendesis anda mungkin mendengar dari sistem audio dengan volume penuh. Untuk gambar digital, noise ini muncul sebagai speckles acak pada permukaan halus dan sebaliknya secara signifikan dapat menurunkan kualitas gambar. Meskipun akan sering mengurangi noise dari sebuah gambar, kadang-kadang noise diinginkan karena dapat menambahkan efek model lama. Beberapa noise dapat juga meningkatkan ketajaman nyata dari suatu gambar. Noise

meningkat dengan pengaturan sensitivitas pada kamera, panjang pemaparan, suhu, dan bahkan bervariasi antara model kamera yang berbeda.



Gambar 2.5 Gambar noise



Gambar 2.6 Gambar setelah mengurangi noise

2.4.3 Pengolahan Citra

Akhir-akhir ini, gambar digital semakin sering digunakan dibandingkan dengan gambar analog. Hal ini dikarenakan kelebihan gambar digital dibandingkan dengan gambar analog, yaitu mudah disimpan, mudah digandakan, mudah dilakukan pengolahan atau manipulasi tanpa harus takut merusak gambar asli. Gonzales (2002) menyatakan bahwa sebuah gambar dapat didefinisikan sebagai fungsi dua dimensi, $f(x,y)$, dimana x dan y adalah koordinat dari gambar dan amplitudo dari f yang dapat disebut intensitas atau gray-level dari sebuah gambar pada titik yang terletak pada

koordinat x dan y . Jika x , y dan amplitudo dari f adalah terbatas dan dapat ditentukan nilainya, maka gambar tersebut adalah gambar digital. Gambar digital terdiri dari kumpulan pixel. Pixel adalah titik yang berisi nilai tertentu yang membentuk sebuah gambar yang lokasinya berada pada koordinat x dan y .

Gonzales (2002) mendefinisikan image processing adalah suatu metode yang digunakan untuk mengolah atau memanipulasi gambar dalam bentuk 2 dimensi. Image processing dapat juga dikatakan segala operasi untuk memperbaiki, menganalisa, atau mengubah suatu gambar. Konsep dasar pemrosesan gambar digital menggunakan image processing diambil dari kemampuan indera penglihatan manusia yang selanjutnya dihubungkan dengan kemampuan otak manusia untuk melakukan proses atau pengolahan terhadap gambar digital tersebut. Dalam sejarahnya, image processing telah diaplikasikan dalam berbagai bentuk, dengan tingkat kesuksesan cukup besar. Seperti berbagai cabang ilmu lainnya, image processing menyangkut pula berbagai gabungan cabang-cabang ilmu, diantaranya optik, elektronik, matematika, fotografi, dan teknologi komputer.

Pada umumnya, objektivitas dari image processing adalah melakukan transformasi atau analisa suatu gambar sehingga informasi baru tentang gambar dibuat lebih jelas.

2.4.3.1 Tahapan-tahapan Pengolahan Citra

Pengolahan citra dapat dilakukan dengan sempurna dan optimal dengan melakukan tahapan-tahapan pengolahan citra yang mengubah citra asli menjadi citra yang mudah dikenali dan diproses oleh komputer.

Tahapan-tahapan yang harus dilakukan sebelum pengolahan citra :

- Mengatur skala abu-abu atau yang sering disebut grayscale.
- Mengatur ambang batas atau yang sering disebut thresholding.
- Mengendalikan dan menghilangkan noise atau gangguan yang mana dapat mengganggu pengolahan citra.
- Melakukan pengaturan kontras atau sering disebut sebagai contrast stretching yang mana berfungsi untuk mengatur pencahayaan pada citra.

2.4.3.1.1 Grayscale

Grayscale adalah proses perubahan nilai pixel dari warna (RGB) menjadi gray-level. Gray-level adalah tingkat warna abu-abu dari sebuah pixel, dapat juga dikatakan tingkat cahaya dari sebuah pixel, menunjukkan tingkat terangnya pixel tersebut dari hitam ke putih. Pada dasarnya, grayscale ini dilakukan untuk meratakan nilai pixel dari tiga nilai RGB menjadi 1 nilai. Untuk memperoleh hasil yang baik dari pixel tidak langsung dibagi menjadi tiga, melainkan terdapat presentasi dari masing-masing nilai tersebut.

Citra grayscale disimpan dalam format 8 bit untuk setiap sample pixel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Pada aplikasi lain seperti pada aplikasi medical imaging dan remote sensing biasa juga digunakan format 10 bit, 12 bit maupun 16 bit.

Jika setiap piksel pada citra kromatik direpresentasikan oleh intensitas R, G, dan B yang masing-masing mewakili nilai untuk warna merah, hijau, dan biru, ada 3 metode

perata-rataan sederhana yang dapat diimplementasikan untuk mengkonversi citra kromatik menjadi citra skala abu:

- Lightness

Metode lightness bekerja dengan merata-ratakan nilai piksel warna yang paling menonjol dan yang paling tidak menonjol. Nilai piksel skala abu dapat dihitung dengan rumus:

$$P_g = \max((R,G,B) + \min(R,G,B))/2 \quad (2.1)$$

- Average

Metode average merata-ratakan dengan sederhana. Metode ini memberikan nilai yang sama pada tiap piksel warna. Nilai piksel skala abu dapat dihitung dengan rumus:

$$P_g = ((R+G+B))/3 \quad (2.2)$$

- Luminosity

Untuk mengubah gambar RGB menjadi gambar grayscale dengan menggunakan rumus (Stephen Gang Wu , 1997, p. 2) dibawah ini :

$$\text{gray} = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (2.3)$$

Keterangan :

R = merah (red)

G = hijau (green)

B = biru (blue)



Gambar 2.7 Gambar asli



Gambar 2.8 Gambar setelah grayscale

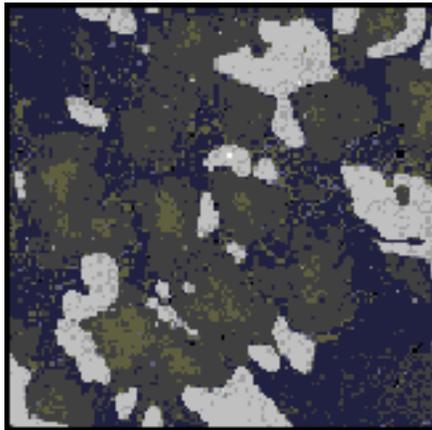
2.4.3.1.2 Thresholding

Gambar hitam putih (binary image) relatif lebih mudah dianalisa dibandingkan dengan gambar berwarna. Karena itu sebelum dianalisa, gambar dikonversikan terlebih dahulu menjadi binary image. Proses konversi ini disebut dengan thresholding. Dalam proses thresholding, warna yang ada dikelompokkan menjadi 0 (hitam) atau 1 (putih). Pengelompokannya berdasarkan pada suatu konstanta ambang batas (level).

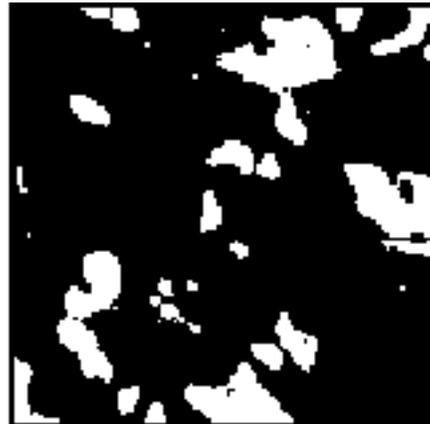
Jika nilai pixel lebih besar sama dengan level, maka nilai output-nya adalah 1 dan sebaliknya, jika nilai pixel lebih kecil dari level, maka nilai output-nya 0. Jika $f(n,m)$ adalah pixel pada gambar awal dan $g(n,m)$ adalah pixel gambar yang sudah melalui proses thresholding, maka :

$$g(n,m) = 0 \quad \text{jika } f(n,m) < \text{level} \quad (2.4)$$

$$g(n,m) = 1 \quad \text{jika } f(n,m) \geq \text{level} \quad (2.5)$$



Gambar 2.9 Gambar asli

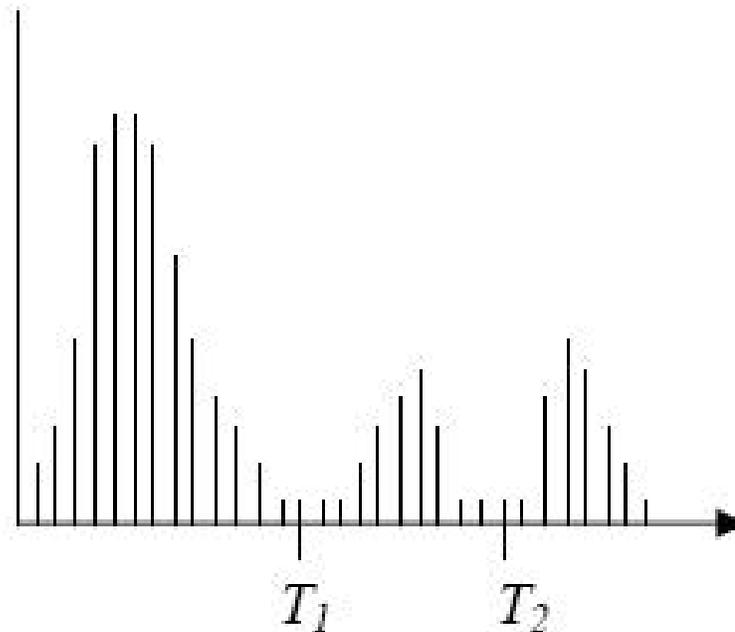


Gambar 2.10 Gambar thresholded

2.4.3.1.2.1 Multilevel Thresholding

Karena graylevel yang terdapat pada citra sangat tipis, maka metode thresholding sederhana tidak mampu menangani citra jenis ini. Sehingga diperlukan suatu metode seperti multilevel thresholding untuk membagi graylevel-graylevel ini menjadi subdaerah yang sesuai.

Multilevel thresholding merupakan proses yang memotong graylevel pada image menjadi beberapa region yang jelas. Teknik thresholding jenis ini membutuhkan lebih dari satu threshold untuk input image dan memotong image menjadi region-region yang pasti. Pada multilevel thresholding, suatu pixel (x,y) dikatakan sebagai satu objek jika $T_1 < f(x,y) \leq T_2$, sebagai objek lain jika $f(x,y) > T_2$, dan sebagai background jika $f(x,y) \leq T_1$.



Gambar 2.11 Multilevel Threshold Histogram.

Terdapat tiga jenis thresholding:

1. Thresholding dikatakan global jika nilai T hanya tergantung pada nilai gray level $f(x,y)$.
2. Thresholding dikatakan local jika nilai T bergantung pada nilai gray level $f(x,y)$ dan nilai properti lokal citra $p(x,y)$.
3. Dan thresholding dikatakan dynamic atau adaptive jika nilai T bergantung pada koordinat spasial x dan y .

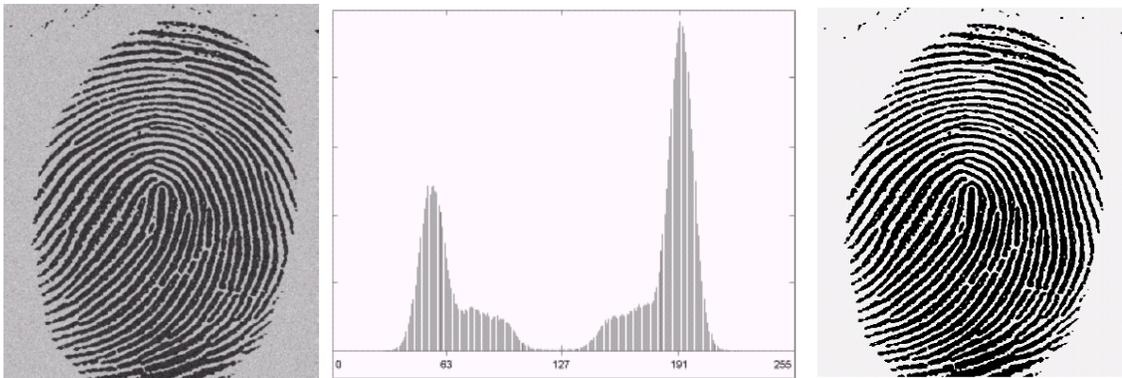
2.4.3.1.2.2 Global Thresholding

Dasar global threshold, T , dihitung dengan langkah-langkah :

1. Pilih perkiraan awal untuk T (biasanya nilai keabuan pada citra)
2. Bagi citra menggunakan T untuk menghasilkan dua anggota piksel : $G1$ mengandung piksel dengan nilai keabuan $>T$ dan $G2$ mengandung piksel dengan nilai keabuan $\leq T$
3. Jumlahkan nilai piksel keabuan pada $G1$ untuk menghasilkan μ_1 dan $G2$ untuk

$$\text{Menghasilkan } \mu_2 : T = \frac{\mu_1 + \mu_2}{2} \quad (2.6)$$

4. Hitung nilai threshold nilai threshold yang baru
5. Ulangi langkah 2 – 4 sampai perbedaan pada T yang berulang-ulang kurang dari batasan T_∞



Gambar 2.12 Thresholding

2.4.3.1.2.3 Local Thresholding

Local thresholding biasa disebut juga dengan adaptive thresholding. Thresholding ini bersifat otomatis dalam penentuan nilai threshold atau nilai ambang batas yang akan digunakan sebagai penunjang besarnya nilai keabuan yang akan muncul.

Pada situasi dimana pencahayaan berada pada ruang dan waktu yang berbeda-beda, sangat dimungkinkan untuk menggunakan local thresholding yang mana dapat beradaptasi untuk merubah kondisi kecerahan dengan membandingkan intensitas relatif dari masing-masing pixel sampai intensitas tetangganya. Untuk setiap pixel, angka komputasi yang mengelilingi piksel dipilih dan setiap algoritma thresholding otomatis dapat digunakan untuk menentukan nilai threshold yang tepat.

Nilai threshold yang dihitung setiap lokasi pikselnya dapat disimpan dalam bentuk array, yang mana dapat digunakan :

- Menganalisa ragam kecerahan
- Menyeimbangkan citra
- Batasan threshold citra

Untuk Menghitung local thresholding dengan fungsi $T(x,y)$:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2.7)$$

2.4.3.1.2.4 Adaptive Thresholding

Nilai threshold sangat mempengaruhi kejelasan objek citra sehingga objek tersebut dapat di proses lebih lanjut. Oleh karena itu nilai threshold yang optimal harus ditentukan agar objek citra menghasilkan citra yang dapat diproses secara maksimum. Ada beberapa metode untuk menentukan nilai threshold maksimum, yaitu metode Otsu, isodata dan convex hull.

- Isodata

Metode ini dikembangkan oleh Ridler dan Calvard. Langkah-langkah metode isodata adalah sebagai berikut :

- Pilih nilai threshold dan nyatakan threshold dengan nilai.
- Bagi histogram menjadi dua bagian sehingga salah satu bagian berhubungan dengan sisi depan dan satunya lagi dengan sisi belakang.
- Hitung nilai uji rata-rata dari nilai keabuan piksel bagian depan dan piksel bagian belakang ($f,0 m$ and $b,0 m$).
- Hitung nilai threshold yang baru $T1$ sebagai rata-rata dari nilai uji sebelumnya.
- Melakukan pembagian kembali histogram menjadi dua bagian.
- Lakukan pengulangan proses berdasarkan threshold baru sampai nilai threshold sesuai. Dengan kata lain

$$T_k = (m_{f,k-1} + m_{b,k-1}) / 2 \text{ until } T_k = T_{k-1} \quad (2.8)$$

- Metode Otsu

Metode ini diperkenalkan oleh Nobuyuki Otsu. Ide utama metode ini adalah mengatur threshold sehingga elemen yang sekelompok saling mendekat satu sama lain.

Metode Otsu mendeskripsikan histogram tingkat keabuan dari sebuah citra sebagai sebuah distribusi probabilitas, sehingga:

$$p_i = n_i / N \quad (2.9)$$

Di mana n_i adalah jumlah piksel dengan nilai keabuan i dan N adalah jumlah total piksel pada citra, sehingga p_i adalah probabilitas dari pixel yang memiliki nilai keabuan i . Jika kita melakukan ambang batas pada level k , kita dapat mendefinisikan:

$$\omega(k) = \sum_{i=0}^k p_i \quad (2.10)$$

$$\mu(k) = \sum_{i=k+1}^{L-1} p_i \quad (2.11)$$

Di mana L adalah jumlah dari derajat keabuan (misalnya 256 untuk citra 8 bit).

Dengan definisi:

$$\omega(k) + \mu(k) = \sum_{i=0}^{L-1} p_i = 1 \quad (2.12)$$

Kita bermaksud akan menemukan k untuk memaksimalkan perbedaan antara $\omega(k)$ dan $\mu(k)$. Hal ini dapat dilakukan pertama-tama dengan mendefinisikan rata-rata nilai derajat keabuan citra dengan:

$$\mu_T = \sum_{i=0}^{L-1} ip_i \quad (2.13)$$

Dan kemudian menemukan nilai k maksimal:

$$\frac{(\mu_T \omega(k) - \mu(k))^2}{\omega(k)\mu(k)} \quad (2.14)$$

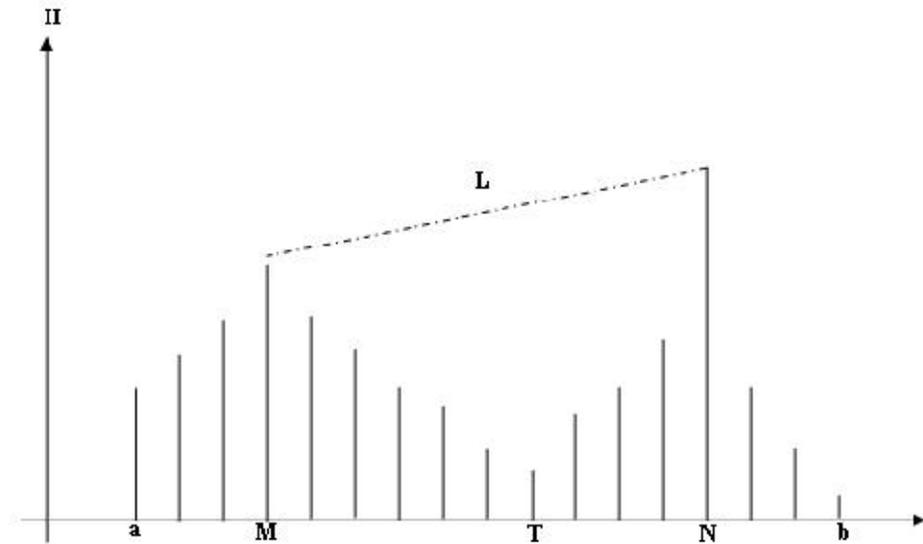
Yang memaksimalkan varians antar kelas (atau meminimalisasi varians di dalam kelas). Nilai k tersebut dipilih untuk memaksimalkan pemisahan antar dua kelas (latar depan dan latar belakang), atau secara alternatif meminimalisasi penyebarannya, sehingga tumpang tindih di antaranya menjadi minimal.

- Convex Hull

Convex hull threshold selection pertama kali dikenalkan oleh Rosenfeld.

Langkah-langkah convex hull :

- Hitung kedua puncak (M,N)
- Hubungkan titik-titik (M, H(M)) dan (N, H(N)) untuk mendapatkan garis L.
- Threshold adalah titik antara [M, N] yang mempunyai jarak maksimum dari garis L



Gambar 2.13 Proses Pengolahan Convex Hull

2.4.3.1.3 Noise Reduction

Smoothing filter digunakan untuk menghaluskan (smoothing) dan mengurangi noise (noise reduction). Smoothing digunakan untuk langkah pre-proses, seperti mengurangi bagian detail yang kecil dari sebuah gambar sebelum ekstraksi objek dan menjembatani dari gap-gap yang kecil dalam garis atau kurva. Noise reduction dapat diselesaikan smoothing dengan filter linear atau dengan filter non-linear.

Rectangular Averaging Filtering merupakan smoothing linear filter. Rectangular Averaging Filtering adalah filter yang digunakan untuk melakukan proses filter noise pada gambar. Rectangular Averaging Filtering biasanya juga disebut Lowpass Spatial Filtering. Sebagai contoh Rectangular Averaging Filtering yang menggunakan template 3x3 dan 5x5 :

$$h_{\text{rect}}[j,k] = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Template 3x3

$$h_{\text{rect}}[j,k] = \frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Template 5x5

Gambar 2.14 Rectangular Averaging Filter

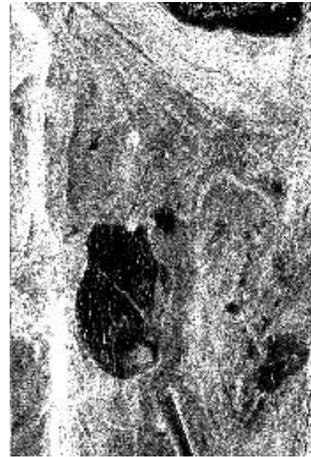
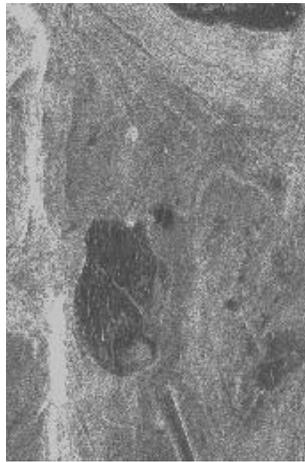
Lowpass Spatial Filtering mempunyai efek untuk melewati atau tidak memroses komponen yang mempunyai frekuensi rendah dari sebuah gambar. Komponen yang berfrekuensi tinggi ditipiskan dan tampak memudar pada gambar hasil. Berikut ini adalah contoh dari proses Rectangular Averaging Filtering :

**Gambar 2.15** Gambar asli**Gambar 2.16** Gambar setelah smoothing

2.4.3.1.4 Contrast Stretching

Contrast Stretching (sering disebut normalisasi) adalah teknik sederhana perbaikan gambar yang mencoba untuk meningkatkan kontras pada foto dengan ‘stretching’ jarak nilai-nilai intensitas berisi renggang yang diinginkan untuk kisaran nilai, misalnya yang jarak penuh nilai-nilai pixel bahwa jenis foto yang bersangkutan memungkinkan. Kontras gambar yang rendah dapat disebabkan oleh iluminasi rendah, kurangnya jangkauan dinamis dalam pencitraan sensor, atau karena pengaturan lensa

yang salah. Gagasan di balik contrast stretching adalah untuk meningkatkan jangkauan dinamis tingkat intensitas dalam gambar diproses.



Gambar 2.17 Gambar asli **Gambar 2.18** Gambar setelah Contrast Stretching

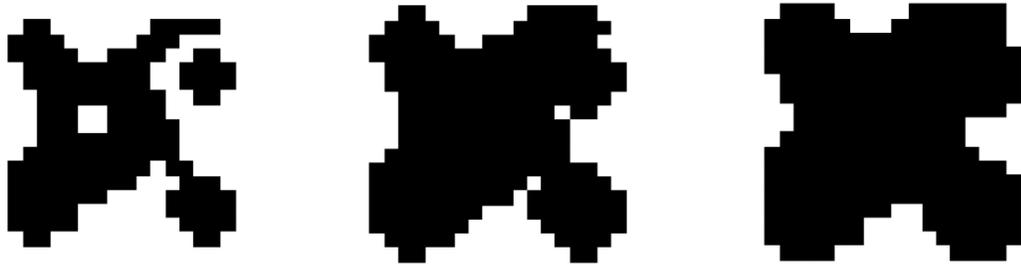
2.4.3.1.5 Morfologi

Pengolahan citra secara morfologi adalah alat untuk mengekstrak atau memodifikasi informasi pada bentuk dan struktur dari objek di dalam citra (Dougherty, 2009). Operator morfologi yang umum digunakan adalah erosi dan dilasi, sedangkan operator lainnya merupakan pengembangan dari keduanya.

2.4.3.1.5.1 Dilasi

Operasi dilasi dilakukan untuk memperbesar ukuran segmen obyek dengan menambah lapisan di sekeliling obyek. Terdapat 2 cara untuk melakukan operasi ini, yaitu dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik obyek, atau lebih mudahnya set setiap titik yang tetangganya adalah titik obyek menjadi titik obyek. Cara kedua yaitu dengan mengubah semua titik di sekeliling titik batas menjadi titik obyek, atau lebih mudahnya set semua titik tetangga sebuah titik

obyek menjadi titik obyek. Contoh citra hasil operasi dilasi diberikan pada gambar 2.17 berikut ini.



(a) Citra asli

(b) Citra dilasi terhubung-4

(c) Citra dilasi terhubung-8

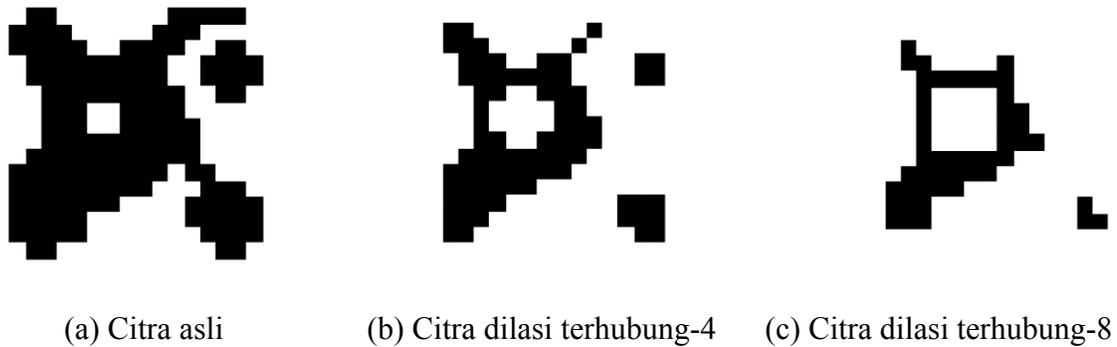
Gambar 2.19 Hasil Dilasi

Algoritma untuk operasi dilasi citra biner :

1. Untuk semua titik dalam citra
2. Cek apakah tersebut titik obyek
3. Jika ya maka ubah semua tetangganya menjadi titik obyek
4. Jika tidak maka lanjutkan

2.4.3.1.5.2 Erosi

Operasi erosi adalah kebalikan dari operasi dilasi. Pada operasi ini, ukuran obyek diperkecil dengan mengikis sekeliling obyek. Cara yang dapat dilakukan juga ada 2. Cara pertama yaitu dengan mengubah semua titik batas menjadi titik latar dan cara kedua dengan menset semua titik di sekeliling titik latar menjadi titik latar. Contoh citra hasil operasi erosi diberikan dalam Gambar 2.20.



Gambar 2.20 Hasil Erosi

Algoritma untuk operasi erosi citra biner :

1. Untuk semua titik dalam citra
2. Cek apakah tersebut titik latar
3. Jika ya maka ubah semua tetangganya menjadi titik latar
4. Jika tidak maka lanjutkan

2.4.3.1.5.3 Penutupan (Closing)

Operasi penutupan adalah kombinasi antara operasi dilasi dan erosi yang dilakukan secara berurutan. Citra asli didilasi terlebih dahulu, kemudian hasilnya dierosi. Contoh hasil operasi penutupan terlihat pada Gambar 2.21. Operasi ini digunakan untuk menutup atau menghilangkan lubang-lubang kecil yang ada dalam segmen obyek. Terlihat pada gambar tersebut, pada citra hasil, tidak terdapat lubang lagi di dalam segmen obyek. Operasi penutupan juga digunakan untuk menggabungkan 2 segmen obyek yang saling berdekatan (menutup sela antara 2 obyek yang sangat berdekatan). Terlihat pada citra asal terdapat 2 buah segmen obyek, namun pada citra hasil hanya tinggal sebuah segmen obyek gabungan. Operasi penutupan dapat juga dilakukan dalam

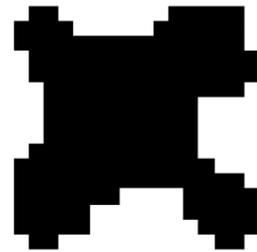
beberapa rangkaian dilasi-erosi (misalnya 3 kali dilasi, lalu 3 kali erosi) apabila ukuran lubang atau jarak antar obyek cukup besar.



(a) Citra asli



(b) Citra closing terhubung-4



(c) Citra closing terhubung-8

Gambar 2.21 Hasil Closing

2.4.3.1.5.4 Pembukaan (Opening)

Operasi pembukaan juga merupakan kombinasi antara operasi erosi dan dilasi yang dilakukan secara berurutan, tetapi citra asli dierosi terlebih dahulu baru kemudian hasilnya didilasi. Contoh hasil operasi pembukaan untuk obyek yang sama terlihat pada Gambar 2.22. Operasi ini digunakan untuk memutus bagian-bagian dari obyek yang hanya terhubung dengan 1 atau 2 buah titik saja. Seperti terlihat, pada citra asal terdapat 2 buah obyek, namun pada citra hasil (b) menjadi 3 segmen karena obyek pertama dipisah menjadi 2. Operasi ini juga digunakan untuk menghilangkan obyek yang sangat kecil. Pada citra hasil (c), obyek kecil di kanan atas tidak muncul lagi.



a) Citra asli (b) Citra opening terhubung-4 (c) Citra opening terhubung-8

Gambar 2.22 Hasil Opening

2.5 Ekstraksi Fitur

Ekstraksi fitur merupakan bentuk khusus dari dimensi yang tereduksi. Ketika data masukkan untuk algoritma begitu besar untuk diproses dan kemungkinan besar informasi data sedikit (banyak data tetapi tidak banyak informasi) kemudian data masukkan akan diubah menjadi penyajian data yang tereduksi yang ditentukan sebagai fitur (disebut juga fitur vektor). Perubahan data masukkan menjadi data yang ditentukan sebagai fitur disebut sebagai ekstraksi fitur. Jika ekstraksi fitur dipilih secara teliti diharapkan data fitur akan mendapatkan informasi yang sesuai dari masukkan data untuk menghasilkan tujuan yang diinginkan menggunakan penyajian data yang telah disaring dari ukuran asli datanya.

Ekstraksi Fitur yang dapat dilakukan terdapat 17 jenis yang terdiri diameter, panjang, lebar, luas serta smooth factor, aspect ratio, form factor, rectangularity, narrow factor, rasio perimeter dari diameter, rasio perimeter.

Menurut (Devaux M-F, Guillemin F, Bouchet B, Guillon F) dua fitur citra sel tumbuhan yang dapat diekstrak dengan menggunakan matlab :

- Ukuran sel tumbuhan

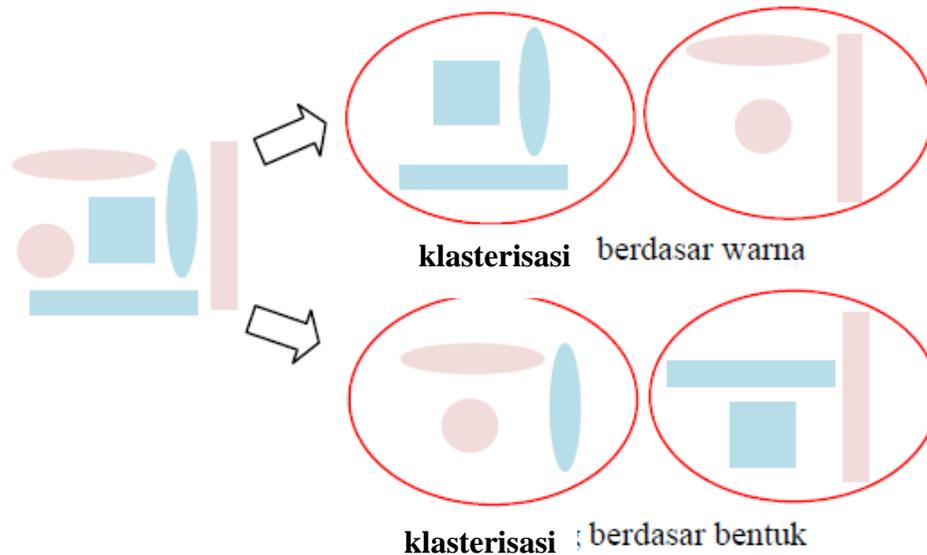
Karena pada citra sel tanaman yang terlihat jelas perbedaan mendasar yaitu ukuran yang berbeda-beda. Dan ukuran pun mempunyai nilai yang bisa dijadikan fitur vektor yang dapat diproses langsung oleh klaster k-means.

- Bentuk sel tumbuhan

Bentuk digunakan karena bentuk merupakan salah satu fitur pada citra sel tanaman yang terlihat perbedaannya. Fitur bentuk paling sering dan umum digunakan pada pengekstraksian fitur pada citra.

2.6 Klasterisasi

Klasterisasi adalah proses mengelompokkan objek berdasarkan informasi yang diperoleh dari data yang menjelaskan hubungan antar objek dengan prinsip untuk memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas atau klaster. Tujuannya menemukan klaster yang berkualitas dalam waktu yang layak. Kesamaan objek biasanya diperoleh dari kedekatan nilai-nilai atribut yang menjelaskan objek-objek data, sedangkan objek-objek data biasanya direpresentasikan sebagai titik dalam ruang multidimensi. Sedangkan klasterisasi image (pengelompokkan gambar) adalah proses untuk membagi atau mengelompokkan suatu image ke dalam beberapa bagian yang berbeda, dimana pada tiap bagian yang berbeda tersebut anggotanya mempunyai kesamaan khusus (homogen). Klasterisasi biasa digunakan pada banyak bidang, seperti : data mining, pattern recognition (pengenalan pola), image classification (pengklasifikasi gambar), ilmu biologi, pemasaran, perencanaan kota, pencarian dokumen, dan lain sebagainya.



Gambar 2.23 Ilustrasi klasterisasi

2.6.1 Karakteristik Klasterisasi

Karakteristik clustering dibagi menjadi 4, yaitu :

1. Klasterisasi partitioning

Klasterisasi Partitioning disebut juga klasterisasi exclusive, dimana setiap data harus termasuk ke klaster tertentu. Karakteristik tipe ini juga memungkinkan bagi setiap data yang termasuk klaster tertentu pada suatu tahapan proses, pada tahapan berikutnya berpindah ke klaster yang lain.

Contoh : K-Means, residual analysis.

2. Klasterisasi hierarchical

Pada klasterisasi hierarchical, Setiap data harus termasuk ke klaster tertentu. Dan suatu data yang termasuk ke klaster tertentu pada suatu tahapan proses, tidak dapat berpindah ke klaster lain pada tahapan berikutnya.

Contoh: Single Linkage, Centroid Linkage, Complete Linkage, Average Linkage.

3. Klasterisasi overlapping

Dalam klasterisasi overlapping, setiap data memungkinkan termasuk ke beberapa klaster. Data mempunyai nilai keanggotaan (membership) pada beberapa klaster.

Contoh: Fuzzy C-means, Gaussian Mixture.

4. Hybrid

Karakteristik hybrid adalah Mengawinkan karakteristik dari partitioning, overlapping dan hierarchical.

2.6.2 Algoritma Klasterisasi

Ada beberapa algoritma yang sering digunakan dalam proses klasterisasi, yaitu :

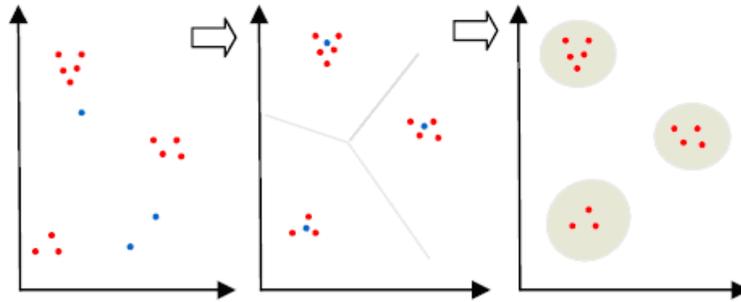
1. K-Means

Termasuk klasterisasi partitioning yang memisahkan data ke k daerah bagian yang terpisah. K-means algorithm sangat terkenal karena kemudahan dan kemampuannya untuk mengklaster data besar dan data outlier dengan sangat cepat. Sesuai dengan karakteristik klasterisasi partitioning, Setiap data harus termasuk ke kelas tertentu, dan Memungkinkan bagi setiap data yang termasuk kelas tertentu pada suatu tahapan proses, pada tahapan berikutnya berpindah ke kelas yang lain.

Algoritma K-Means :

1. Menentukan k sebagai jumlah kelas yang ingin dibentuk
2. Membangkitkan k centroids (titik pusat kelas) awal secara random
3. Menghitung jarak setiap data ke masing-masing centroids
4. Setiap data memilih centroids yang terdekat

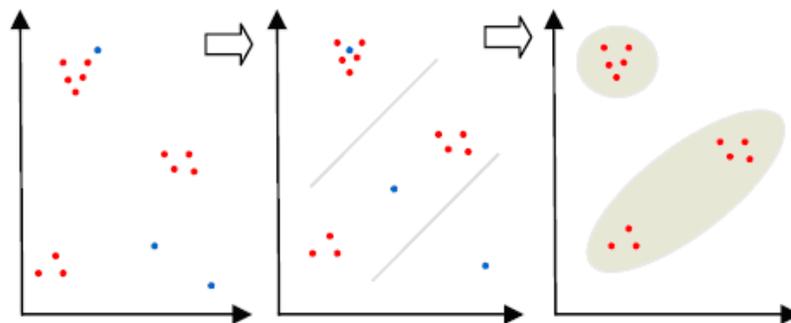
5. Menentukan posisi centroids baru dengan cara menghitung nilai rata-rata dari data-data yang memilih pada centroid yang sama.
6. Kembali ke langkah 3 jika posisi centroids baru dengan centroids lama tidak sama.



Gambar 2.24 Ilustrasi Algoritma K-Means

Karakteristik K-Means :

- K-Means sangat cepat dalam proses klusterisasi
- K-Means sangat sensitif pada pembangkitan centroids awal secara random
- Memungkinkan suatu kluster tidak mempunyai anggota



Gambar 2.25 Ilustrasi kelemahan K-Means

2. Klasterisasi Hierarchical

Dengan metode ini, data tidak langsung dikelompokkan kedalam beberapa klaster dalam 1 tahap, tetapi dimulai dari 1 klaster yang mempunyai kesamaan, dan berjalan seterusnya selama beberapa iterasi, hingga terbentuk beberapa klaster tertentu.

Arah klasterisasi hierarchical dibagi 2, yaitu :

a) Divisive

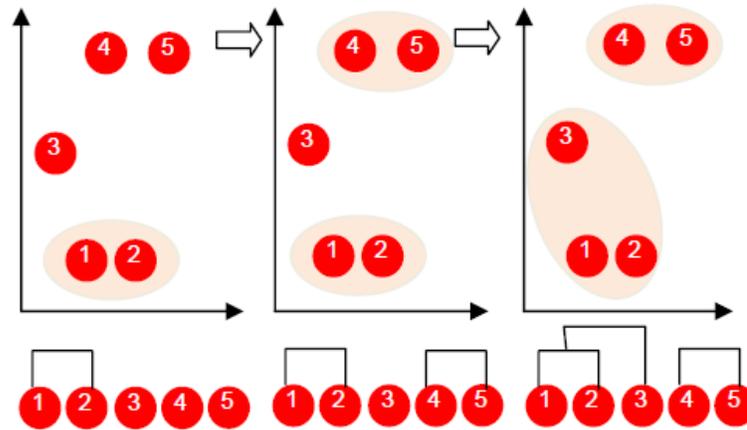
- Dari 1 klaster ke k klaster
- Pembagian dari atas ke bawah (top to down division)

b) Agglomerative

- Dari N klaster ke k klaster
- Penggabungan dari bawah ke atas (down to top merge).

Algoritma klasterisasi hierarchical :

1. Menentukan k sebagai jumlah klaster yang ingin dibentuk
2. Setiap data dianggap sebagai klaster. Kalau $N =$ jumlah data dan $c =$ jumlah klaster, berarti ada $c = N$.
3. Menghitung jarak antar klaster
4. Cari 2 klaster yang mempunyai jarak antar klaster yang paling minimal dan gabungkan (berarti $c = c - 1$).
5. Jika $c > k$, kembali ke langkah 3.



Gambar 2.26 Ilustrasi Algoritma Klasterisasi Hierarchical

Penghitungan jarak antar obyek, maupun antar klasternya dilakukan dengan Euclidian distance, khususnya untuk data numerik. Untuk data 2 dimensi, digunakan persamaan sebagai berikut :

$$d(x,y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad (2.15)$$

Algoritma klasterisasi hierarchial banyak diaplikasikan pada metode pengklasteran berikut :

1. Single Linkage Hierarchical Method (SLHM)

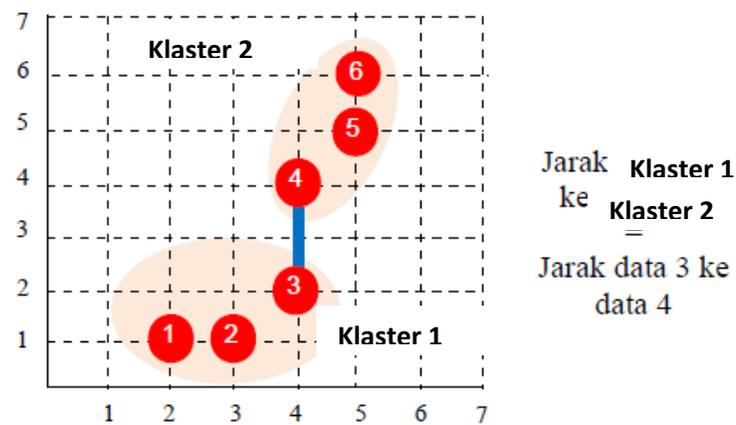
Single Linkage adalah proses pengklasteran yang didasarkan pada jarak terdekat antar obyeknya (minimum distance).

Metode SLHM sangat bagus untuk melakukan analisa pada tiap tahap pembentukan klaster. Metode ini juga sangat cocok untuk dipakai pada kasus klasterisasi shape independent, karena kemampuannya

untuk membentuk pattern/pola tertentu dari klaster. Sedangkan untuk kasus klasterisasi condensed, metode ini tidak bagus.

Algoritma Single Linkage Hierarchical Method :

1. Diasumsikan setiap data dianggap sebagai klaster. Kalau n =jumlah data dan c =jumlah klaster, berarti ada $c=n$.
2. Menghitung jarak antar klaster dengan Euclidian distance.
3. Mencari 2 klaster yang mempunyai jarak antar klaster yang paling minimal dan digabungkan (merge) kedalam klaster baru (sehingga $c=c-1$)
4. Kembali ke langkah 3, dan diulangi sampai dicapai klaster yang diinginkan.



Gambar 2.27 Ilustrasi Single Linkage

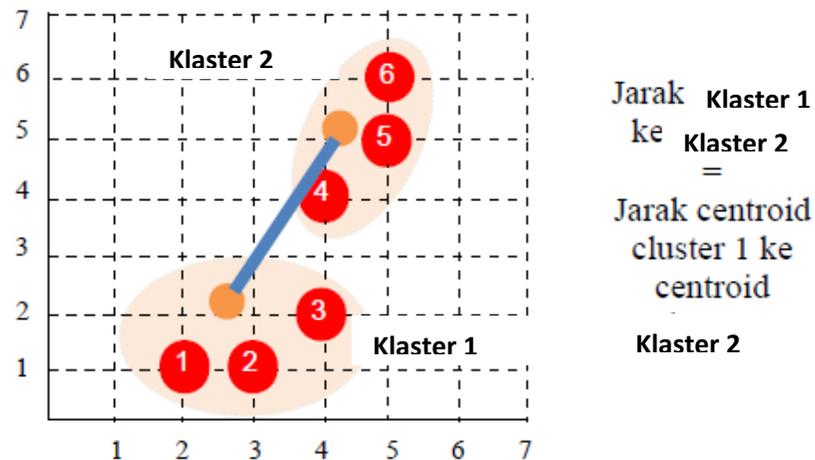
2. Centroid Linkage Hierarchical Method

Centroid Linkage adalah proses pengklasteran yang didasarkan pada jarak antar centroidnya. Metode ini bagus untuk memperkecil variance within klaster karena melibatkan centroid pada saat

penggabungan antar kluster. Metode ini juga baik untuk data yang mengandung outlier.

Algoritma Centroid Linkage Hierarchical Method :

- i. Diasumsikan setiap data dianggap sebagai kluster. Kalau n =jumlah data dan c =jumlah kluster, berarti ada $c=n$.
- ii. Menghitung jarak antar kluster dengan Euclidian distance.
- iii. Mencari 2 kluster yang mempunyai jarak centroid antar kluster yang paling minimal dan digabungkan (merge) kedalam cluster baru (sehingga $c=c-1$)
- iv. Kembali ke langkah 3, dan diulangi sampai dicapai kluster yang diinginkan.



Gambar 2.28 Ilustrasi Centroid Linkage

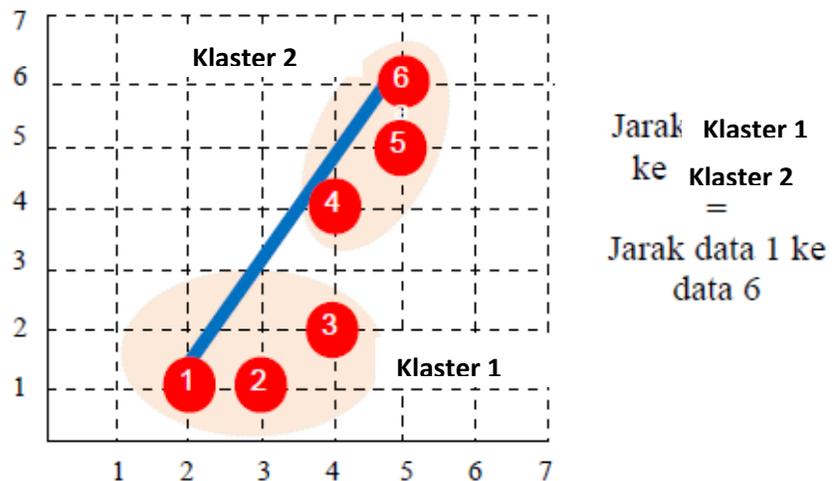
3. Complete Linkage Hierarchical Method

Complete Linkage adalah proses pengklasteran yang didasarkan pada jarak terjauh antar obyeknya (maksimum distance). Metode ini baik

untuk kasus klasterisasi dengan normal data set distribution. Akan tetapi, metode ini tidak cocok untuk data yang mengandung outlier.

Algoritma Complete Linkage Hierarchical Method :

1. Diasumsikan setiap data dianggap sebagai klaster. Kalau n =jumlah data dan c =jumlah klaster, berarti ada $c=n$.
2. Menghitung jarak antar klaster dengan Euclidian distance.
3. Mencari 2 cluster yang mempunyai jarak antar klaster yang paling maksimal atau terjauh dan digabungkan (merge) kedalam klaster baru (sehingga $c=c-1$)
4. Kembali ke langkah 3, dan diulangi sampai dicapai klaster yang diinginkan.

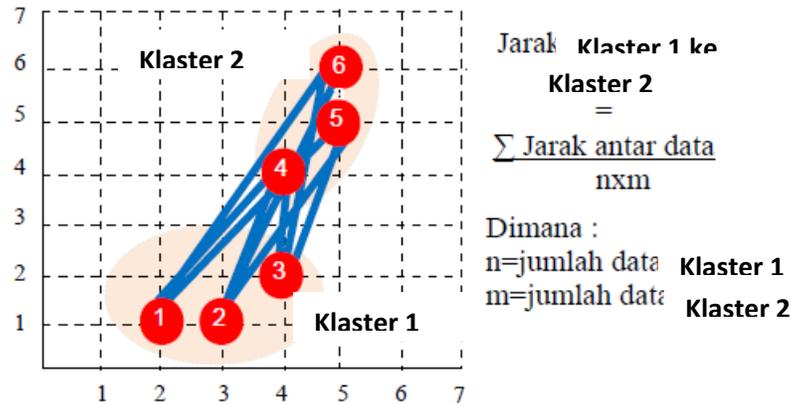


Gambar 2.29 Ilustrasi Complete Linkage

4. Average Linkage Hierarchical Method

Average Linkage adalah proses pengklasteran yang didasarkan pada jarak rata-rata antar obyeknya (average distance). Metode ini relatif

yang terbaik dari metode-metode hierarchical. Namun, ini harus dibayar dengan waktu komputasi yang paling tinggi dibandingkan dengan metode-metode hierarchical yang lain.



Gambar 2.30 Ilustrasi Average Linkage

2.6.3 Analisa Klasterisasi

Analisa klaster adalah suatu teknik analisa multivariate (banyak variabel) untuk mencari dan mengorganisir informasi tentang variabel tersebut sehingga secara relatif dapat dikelompokkan dalam bentuk yang homogen dalam sebuah klaster. Secara umum, bisa dikatakan sebagai proses menganalisa baik tidaknya suatu proses pembentukan klaster. Analisa klaster bisa diperoleh dari kepadatan klaster yang dibentuk (klaster density). Kepadatan suatu klaster bisa ditentukan dengan variance within klaster (V_w) dan variance between cluster (V_b).

Varian tiap tahap pembentukan cluster bisa dihitung dengan rumus :

$$V_C^2 = \frac{1}{n_c - 1} \sum_{i=1}^n (y_i - y_c)^2 \quad (2.16)$$

Dimana,

V_c^2 = varian pada klaster c

c = 1..k, dimana k = jumlah klaster

n_c = jumlah data pada klaster c

y_i = data ke-i pada suatu klaster y_i = rata-rata dari data pada suatu klaster

Selanjutnya dari nilai varian diatas, kita bisa menghitung nilai variance within klaster

(Vw) dengan rumus :

$$V_w = \frac{1}{N-c} \sum_{i=1}^c (n_i - 1) \cdot V_i^2 \quad (2.17)$$

Dimana,

N = Jumlah semua data

n_i = Jumlah data cluster i

V_i = Varian pada cluster i

Dan nilai variance between cluster (Vb) dengan rumus :

$$V_b = \frac{1}{c-1} \sum_{i=1}^c n_i (y_i - \bar{y})^2 \quad (2.18)$$

Dimana,

\bar{y} = rata-rata dari y_i

Salah satu metode yang digunakan untuk menentukan klaster yang ideal adalah batasan variance, yaitu dengan menghitung kepadatan klaster berupa variance within klaster (Vw) dan variance between klaster (Vb). Kluster yang ideal mempunyai Vw minimum yang merepresentasikan internal homogeneity dan maksimum Vb yang menyatakan external homogeneity.

$$V = \frac{V_w}{V_b} \quad (2.19)$$

Meskipun minimum V_w menunjukkan nilai klaster yang ideal, tetapi pada beberapa kasus kita tidak bisa menggunakannya secara langsung untuk mencapai global optimum. Jika kita paksakan, maka solusi yang dihasilkan akan jatuh pada local optima.

Adapun metode-metode dalam analisa klaster yang berguna untuk menentukan optimisasi jumlah klaster yang akan dibentuk yaitu Hartigan, Krzanowski-Lai, silhouette dan gap statistic. Berdasarkan (Prof. Dra. Susanti Linuwih, M.Stats., Ph.D) yang menghasilkan nilai optimum paling baik adalah dengan metode Silhouette dan gap statistic.

2.6.4 Penentuan Jumlah Kelas yang Optimal

2.6.4.1 Gap Statistic

The gap statistic adalah metode untuk menentukan banyaknya kelas yang optimum untuk dibentuk. Teknik ini berdasarkan ide untuk mengubah perubahan kelas dengan meningkatkan nilai klaster yang diperkirakan. Data diambil secara acak. Pertama, kita asumsikan ada sebuah contoh data (x_i) yang lalu digunakan oleh metode klaster, Nilai resultan dari klaster C_1, C_2, \dots, C_k dapat diperoleh untuk klaster C_r , jumlah kedua jarak $d^2(x_i, x_{i'})$, untuk semua titik pada klaster r di hitung. Dan jumlahkan perubahan klaster W_k didefinisikan dalam perhitungan berikut :

$$W_k = \sum_{r=1}^k \frac{1}{2n_r} \sum_{i, i' \in C_r} d^2(x_i, x_{i'}) \quad (2.20)$$

Sebagai konsep utama dari gap statistic, hal ini untuk membandingkan $\log(W_k)$ dengan perkiraan dalam kisaran yang ada. Didefinisikan :

$$Gap_n(k) = E_n^* \{ \log(W_k) \} - \log(W_k) \quad (2.21)$$

Yang mana En^* menunjuk kepada contoh-contoh perkiraan ukuran. Optimalisasi kelas k ditentukan $Gapn(k)$. Pada kasus ini, memodifikasi gap statistic dengan merubah W_k untuk digunakan pada kluster K-Means.

2.6.4.2 Silhouette

Asumsikan data yang telah berkumpul teknik melalui apapun, seperti K-Means, ke dalam k kluster. Untuk setiap data, i membiarkan $a(i)$ menjadi rata-rata ketidaksamaan i dengan semua data di dalam kluster yang sama. Setiap ukuran ketidaksamaan dapat digunakan, tetapi jarak adalah yang paling umum. Kita bisa menafsirkan sebuah $a(i)$ sebagai i cocok seberapa baik adalah kluster itu ditetapkan (nilai yang lebih kecil, lebih baik yang cocok). Kemudian Menemukan ketidaksamaan dari i rata-rata data tunggal dengan kluster lain. Ulangi langkah ini untuk setiap kluster yang i bukan sebagai anggota. Menunjukkan kluster dengan rata-rata terendah ketidaksamaan untuk i oleh $b(i)$. Kluster ini dikatakan menjadi kelompok tetangga i sebagaimana adanya, selain dari kluster i ditetapkan, kluster terbaik i cocok masuk kita sekarang mendefinisikan: pulsa kelompok ini dikatakan tetangga saya sebagaimana adanya, terlepas dari kluster i ditetapkan, kluster i sangat cocok. Kita sekarang definisikan:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (2.22)$$

Dapat ditulis sebagai:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & \text{if } a(i) > b(i) \end{cases} \quad (2.23)$$

Dari definisi di atas jelas bahwa

$$-1 \leq s(i) \leq 1 \quad (2.24)$$

Untuk $s(i)$ untuk dekat dengan 1 kita memerlukan $a(i) \ll b(i)$. Sebagai $a(i)$ adalah ukuran bagaimana i berbeda dengan kluster sendiri, nilai kecil artinya serasi. Selain itu, besar $b(i)$ menyiratkan bahwa i sangat cocok untuk kluster tetangganya. Jadi sebuah $s(i)$ yang dekat dengan salah satu berarti bahwa data yang tepat berkumpul. Jika $s(i)$ dekat dengan negatif, maka dengan logika yang sama kita melihat bahwa i akan lebih tepat kalau itu berkumpul dalam kluster tetangganya. Sebuah $s(i)$ mendekati nol berarti bahwa data berada di perbatasan dua kelompok.

Rata-rata $s(i)$ dari sebuah kluster adalah ukuran dari bagaimana erat dikelompokkan semua data di kluster tersebut. Jadi rata-rata $s(i)$ seluruh dataset adalah ukuran dari seberapa tepat data yang telah dikumpulkan. Jika terlalu banyak atau terlalu sedikit kluster, seperti pilihan k yang buruk dalam algoritma K-Means, beberapa kelompok akan menampilkan silhouette jauh lebih sempit dari yang lain. Jadi silhouette plot dan rata-rata adalah alat yang ampuh untuk menentukan jumlah kluster alam dalam dataset.