BAB II
LANDASAN TEORI

2.1 *Information Retrieval*

"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."(Manning, 2008)

Informasi atau data yang dicari dapat berupa berupa teks, image, audio, video dan lain-lain. Koleksi data teks yang dapat dijadikan sumber pencarian juga dapat berupa pesan teks, seperti e-mail, fax, dan dokumen berita, bahkan dokumen yang beredar di internet. Dengan kata lain jumlah dokumen koleksi yang besar sebagai sumber pencarian, maka dibutuhkan suatu sistem yang dapat membantu user menemukan dokumen yang relevan dalam waktu yang singkat dan tepat.

Di teknologi informasi terdapat istilah data *retrieval*, selain *information retrieval*. Dua hal ini sangatlah berbeda. Data *retrieval* secara umum menentukan dokumen yang tepat dari suatu koleksi data, yang isi dokumen tersebut mengandung keyword di dalam query user, namun tidak akan pernah cukup untuk memenuhi kebutuhan informasi *user*. Berbeda dengan data *retrieval*, *user* dari sistem *Information Retrieval* lebih memperhatikan dalam mendapatkan (*retrieve*) informasi melalui subyek, dari pada *retrieve* data berdasarkan query yang diberikan, karena user tidak mengetahui bagaimana proses yang sedang berlangsung.

Tabel 2.1 Perbedaan Information Retrieval dan Data Retrieval

Information Retrieval	**Data *Retrieval***
Berhubungan dengan text bahasa umum yang tidak selalu terstruktur dan ada kemungkinan memiliki kerancuan arti	Berhubungan dengan data, yang mana semantik strukturnya sudah terdefinisikan
Informasi yang diambil mengenai subyek atau topik	Isi dokumen/data mengandung bagian dari keyword

Semantik sering kali hilang	Semantik terdefinisi dengan baik
Kesalahan kecil masih bisa ditorensi	Kesalahan kecil/tunggal dari suatu obyek menunjukkan kegagalan

Menurut Manning, 2008 model yang terdapat dalam *Information Retrieval* terbagi dalam beberapa model besar, yaitu:

1. *Booelan Retrieval*. Contoh model ini ialah *standard Boolean model* dan *extended Boolean model*.
2. *The term vocabulary and postings lists.*
3. *Dictionaries and tolerant retrieval.*
4. *Index construction.*
5. *Index compression.*
6. *Scoring, term weighting, and the vector space model*, model merepresentasikan dokumen dan *query* sebagai vektor atau matriks *similarity* antara vektor dokumen dan vektor *query* yang direpresentasikan sebagai sebuah nilai skalar. Contoh model ini ialah *vector space model* dan *latent semantic indexing* (LSI).
7. *Computing scores in a complete search system.*
8. *Evaluation in information retrieval.*
9. *Relevance feedback and query expansion.*
10. *XML retrieval.*
11. *Probabilistic information retrieval,* model memperlakukan proses pengembalian dokumen sebagai sebuah *probabilistic inference*. Contoh model ini ialah penerapan teorema bayes dalam model probabilistik.
12. *Language models for information retrieval.*
13. *Text classification and Naive Bayes.*
14. *Vector space classification.*
15. *Support vector machines and machine learning on documents.*
16. *Flat clustering.*
17. *Hierarchical clustering.*
18. *Matrix decompositions and latent semantic indexing.*

19. *Web search basics.*
20. *Web crawling and indexes.*
21. *Link analysis.*

Proses dalam *Information Retrieval* dapat digambarkan sebagai sebuah proses untuk mendapatkan *relevant documents* dari *collection documents* yang ada melalui pencarian *query* yang diinputkan user.

Proses yang terjadi di dalam *Information Retrieval System* terdiri dari 2 bagian utama, yaitu *Indexing subsystem*, dan *Searching subsystem* (*matching system*). Proses *indexing* dilakukan untuk membentuk basisdata terhadap koleksi dokumen yang dimasukkan, atau dengan kata lain, *indexing* merupakan proses persiapan yang dilakukan terhadap dokumen sehingga dokumen siap untuk diproses. Proses *indexing* sendiri meliputi 2 proses, yaitu *document indexing* dan *term indexing*. Dari *term indexing* akan dihasilkan koleksi kata yang akan digunakan untuk meningkatkan performansi pencarian pada tahap selanjutnya. Tahap-tahap yang terjadi pada proses *indexing* ialah :

1. *Word Token*

   Yaitu mengubah dokumen menjadi kumpulan *term* dengan cara menghapus semua karakter dalam tanda baca yang terdapat pada dokumen dan mengubah kumpulan *term* menjadi *lowercase*.
2. *Stopword Removal*

   Proses penghapusan kata-kata yang sering ditampilkan dalam dokumen seperti: *and, or, not* dan sebagainya.
3. *Stemming*

   Proses mengubah suatu kata bentukan menjadi kata dasar.
4. *Term Weighting*

   Proses pembobotan setiap *term* di dalam dokumen.

Search subsystem (matching) merupakan proses menemukan kembali informasi (dokumen) yang relevan terhadap *query* yang diberikan. Tidak semua dokumen yang diambil (*retrieved*) oleh sistem merupakan dokumen yang sesuai dengan keinginan *user* (*relevant*).

2.2 *Music Information Retrieval*

"*Music Information Retrieval (MIR) is looking into describing the bits of the digital music in ways that facilitate searching through this abundant world without structure."* (Jehan, 2005) Beberapa topik yang paling terpopuler di bidang *Music Information Retrieval* adalah :

A. *Fingerprinting* : bertujuan menggambarkan permukaan audio lagu dengan representasi yang rapi secara simetris dari beberapa lagu yang disebut *musical signature*. Menurut Cano didalam penelitian Jehan " *The technology enables, for example, cell-phone carriers or copyright management services to automatically identify audio by comparing unique "fingerprints" extracted live from the audio with fingerprints in aspecially compiled music database running on a central server*." (Jehan, 2005)

B. *Query by description* : Terdiri dari sebuah MIDI yang besar atau database suara yang disediakan dalam bentuk kualitatif deskripsi teks dari musik atau "irama" yang di mainkan dari sebuah lagu yang masuk didalam microphone. Menurut B. Whitman and R. Rifkin didalam penelitian Jehan "*The system typically compares the entry with a pre-analyzed database metric, and usually ranks the results by similarity."* (Jehan, 2005)

C. *Music similarity* adalah percobaan mengkalkulasikan sinyal dari musik. Menurut M. Welsh didalam penelitian Jehan, "*There are many criteria with which we may estimate similarities, including editorial (title, artist, country), cultural (genre, subjective qualifiers), symbolic (melody, harmony, structure), perceptual (energy, texture, beat), and even cognitive (experience, reference).*" (Jehan, 2005)

D. *Classification* Menurut L. Lu, H. Jiang, dan H.-J. Zhang didalam penelitian Jehan mengatakan, "*tasks integrate similarity technologies as a way to cluster music into a finite set of classes, including genre, artist, rhythm, instrument, etc*." (Jehan, 2005) Kemiripan dan klasifikasi seringkali tampak paling utama dari cara

mendefenisikan “kebenaran” untuk memberi sebagai fakta tentang perubahan hasil tanpa ada kesalahan.

E. *Thumbnailing* bertujuan untuk membangun “*representative*” kesimpulan suara dari beberapa musik, seperti halya memindahkan yang paling banyak dan menonjolkan yang paling sedikit dari file tersebut. Menurut Peeters didalam penelitian Jehan, “*The task is to detect the boundaries and similarities of large musical structures, such as verses and choruses, and finally assemble them together*.” (Jehan, 2005)

2.2.1 Pengertian Musik

Menurut kamus besar bahasa Indonesia, musik adalah suara yang disusun demikian rupa sehingga mengandung irama, lagu dan keharmonisan terutama suara yang dihasilkan dari alat-alat yang dapat menghasilkan irama walaupun musik adalah sejenis fenomena intuisi, intuk mencipta, memperbaiki dan mempersembahkannya adalah suatu bentuk seni, mendengar musik pula adalah sejenis hiburan, musik adalah sebuah fenomena yang sangat unik yang bisa dihasilkan oleh beberapa alat musik.

2.2.2 Pengertian Lagu

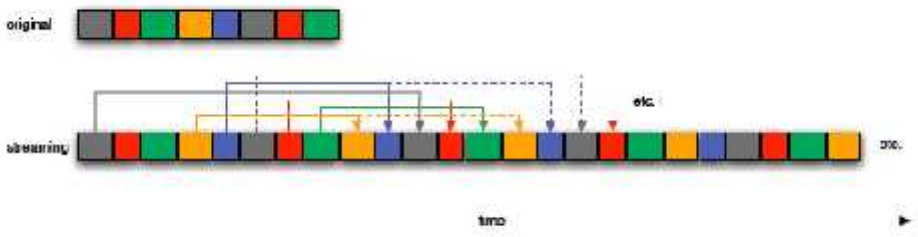
Menurut kamus besar bahasa Indonesia, lagu merupakan gubahan seni atau suara dalam urutan, kombinasi dan hubungan temporal (biasanya diiringi dengan alat musik) untuk menghasilkan gubahan musik yang mempuyai kesatuan dan kesinambungan (mengandung irama).

2.2.3 Pengenalan Tekstur Musik

Dalam kenyataanya bahwa didalam pengumpulan musik secara manual tidak hanya mengumpulkan kembali musik yang sudah lama tetapi juga harus mengumpulkan musik yang baru. Ini adalah masalah yang sangat kompleks yang harus di atasi dari waktu dan hirarki yang harus dihitung dari parameter. Menurut Pachet didalam penelitian Jehan mengatakan, “*The system that probably is the closest to achieving this task is Francois Pachet’s “Continuator”, based on a*

structural organization of Markov chains of MIDI parameters: a kind of prefix tree, where each node contains the result of a reduction function and a list of continuation indexes.(Jehan, 2005)

Memberikan sebuah potongan lagu yang pendek, kita dapat menghasilkan atau menemukan sebuah "*Infinite"* versi lengkap dari musik dengan beberapa tempo yang identik, beberapa kemiripan suara, namun masih terjadi kerancuan. Kita bisa memanggilnya dengan sebutan "tekstur musik". Menurut Schodl didalam penelitian Jehan , *"A variant of this called "audio texture," also inspired by is proposed at the frame level in for textural sound effects (e.g., rain, water stream, horse neighing), i.e., where no particular temporal structure is found."*(Jehan, 2005)



Gambar 2.1 Bagan prosedur dari tekstur musik

(Sumber : Jehan, 2005)

Keterangan :

Energi Fitur Key Tempo

Loudness Mode

2.2.4 Proses Identifikasi Musik

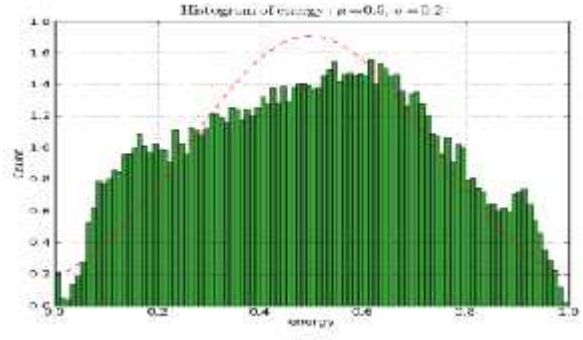
Identifikasi musik dalam penelitian ini dengan menggunakan beberapa fitur yang telah di tetapkan yaitu Energi, *Key*, *Loudness*, *Mode*, dan Tempo.

A. Energi

Ditinjau dari perspektif fisika, setiap sistem fisik mengandung (secara alternatif, menyimpan) sejumlah energi berapa tepatnya ditentukan dengan mengambil jumlah dari sejumlah persamaan khusus, masing-masing didesain untuk mengukur energi yang disimpan secara khusus. Secara umum, adanya energi diketahui oleh pengamat setiap ada pergantian objek atau sistem.

Menurut Budd didalam buku Deutsch mengatakan, "*The fact that music can evoke strong emotions is a mystery that has fascinated scholars since ancient Greece*."(Deutsch, 2013) Pertanyaan-pertanyaan tentang musik dan emosi ada pada perasaan masing masing, ketika kita mendengar musik bagaimana efeknya kepada kita dan sebaliknya bagaimana sikap kita terhadap efek musik tersebut. Namun menurut Pinker didalam buku Deutsch, " *On the one hand, we have "music," an abstract form of art that seems distant from our concerns in everyday life and is commonly regarded as a harmless leisure activity.* Serta menurut Plutchik didalam buku Deutsch, "*On the other hand, we have "emotions," evolved mechanisms that have served important functions in human survival throughout evolution."*(Deutsch, 2013)

Namun pengertian Energi dari musik yang dimiliki oleh sebuah lagu adalah suatu fitur lagu yang merepresentasikan suatu tingkat dari kemampuan suatu musik untuk meningkatkan emosi dari pendengarnya. Artinya pada satu musik dapat memiliki energi yang meningkat, namun pada musik lain memiliki energi yang cenderung menurun. Kontrol pada energi dapat dilakukan dengan meningkatkan jumlah instrument, mengatur dinamika, dan irama pada musik tersebut. Semakin energik suatu musik, maka nilai energi akan semakin besar, begitu pula sebaliknya, semakin lembut suatu musik, maka semakin kecil nilai energinya. (Astawa, 2012)



Gambar 2.2 Kurva *Energy* yang telah di analisis Echonest.
(Sumber : Jehan, 2010)

Nilai *energy* dari sebuah lagu ditentukan dengan menggabungkan nilai dari beberapa fitur lagu seperti *loudness* dan *segment* lagu. Gambar 2.2 diatas adalah nilai *energy* lagu yang telah diekstraksi Echo Nest.

B. *Fitur key*

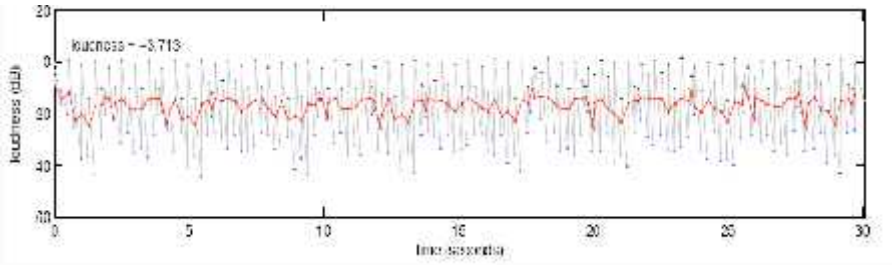
Fitur Key adalah nada dasar yang diperkirakan dari suatu musik. Analisis *Key* dari suatu lagu didasarkan pada nada *tonic triad* serta penggunaan akord *mayor* atau *minor*. Key hasil ekstrasi musik bernilai 0 sampai dengan 11.

C. Loudness

Loudness memiliki defenisi sebagai gambaran kekuatan dari range suara dari yang paling lemah hingga yang paling kuat. Menurut Schraf didalam buku Florentine mengatakan bahwa, *"as the attribute of a sound that changes most readily when sound intensity is varied, but preferred to define it as the subjective intensity of a sound."* (Florentine, 2010). Istilah "subjektif" memberikan perubahan intensitas dalam mendengar. Untuk itu "benar dan salah" adalah jawaban yang sering didengar didalam *loudness.* Ada tidaknya objektif didalam ukuran *loudness*, beberapa ukuran dari loudness harus menggunakan beberapa metode dalam mengumpulkan fakta-fakta. Beberapa defenisi dari loudness merupakan samar-samar tetapi banyak orang memiliki pendapat sendiri cara menentukan ukuran *loudness*.

Loudness merupakan tingkat kenyaringan dari suatu musik dengan satuan *decibels* (dB), di mana *loudness* merupakan bagian dari segmen (satu entitas suara di bawah satu detik) yang nilainya diperoleh dari dua data yaitu nilai dB pada *onset* dan nilai dB maksimum pada musik. *Loudness* juga merupakan suatu nilai yang

menyatakan kualitas dari suatu musik. (Astawa, 2012)



Gambar 2.3 Kurva *Loudness*
(Sumber : Jehan, 2010)

Dalam gambar 2.3, kurva *loudness* digunakan untuk menggambarkan nilai dari *loudness* sebuah lagu. Dalam kurva *loudness* terdapat poin yang menunjukkan nilai dalam satuan dB dan nilai dB maksimum dari suatu lagu. Untuk menentukan nilai loudness didalam echonest adalah :

$$L_{db}(t) = \frac{\sum_{k=1}^{N} Ek(t)}{N} \qquad (2.1)$$

Keterangan :

L_{db} (t) = Loudness
N = Auditory Spectogram
k = Total
Ek = Amplitudo dari Frekuensi
t = waktu

D. *Mode*

Mode adalah nilai akord secara umum dengan nilai "1" untuk *mayor* dan nilai "0" untuk *minor* yang menjadi skala yang terkait dengan not/nada, hasil ekstraksi *mode* memiliki nilai 0 atau 1.

E. Tempo

Fitur lain yang digunakan adalah tempo. Menurut Chang didalam penelitian Astawa, "Tempo adalah kecepatan rata-rata musik dengan nilai *beats per minutes* (BPM)." *Beats* merupakan unit dasar waktu dari potongan musik yang dihitung dalam detik. Kecepatan rata-rata lagu yang cepat jatuh pada range 140 - 200 bpm (*allegro, vivace, presto*) sedangkan yang lambat memiliki tempo di antara 40 dan 280 bpm (*largo, lento, adagio*).

2.3 Echonest

Echonest adalah adalah suatu perusahaan musik yang menyediakan *web service* tentang musik. Ribuan aplikasi berbasis musik dibangun dalam Echonest *Platform*. Echonest menyediakan API *Key* yang bisa digunakan untuk mendapatkan data dari web Echonest secara gratis (Jehan, 2010). API *Key* adalah kode unik sepanjang 12 digit yang dikeluarkan oleh Echonest untuk setiap pengguna Echonest. Kode ini digunakan untuk mengakses beberapa layanan yang dikeluarkan Echonest atau sebagai *database* dari aplikasi yang akan dibangun nantinya. Salah satu contoh penggunaan Echonest API adalah untuk menganalisa *file* *.mp3. Seperti gambar dibawah ini :

Gambar 2.4 Api Key dari Echonest

(Sumber : Echonest, 2013)

Output data yang dikeluarkan oleh Echonest berupa (Jehan,2010) :

a. ***meta data** : analyze, compute, and track information.*

b. ***track data*** *:*
    1. *time signature : an estimated overall time signature of a track.*
    2. *Key : the estimated overall key of a track.*
    3. *Mode : indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived.*
    4. *Tempo : the overall estimated tempo of a track in beats per minute (BPM).*
    5. *Loudness : the overall loudness of a track in decibels (dB).*
    6. *Duration : the duration of a track in seconds as precisely computed by the audio decoder.*
    7. *end of fade in : the end of the fade-in introduction to a track in seconds.*
    8. *start of fade out : the start of the fade out at the end of a track in seconds.*
    9. *codestring, echoprintstring: these represent two different audio fingerprints computed on the audio and are used by other Echo Nest services for song identification.*
c. ***timbre, pitch, and loudness*** *are described in detail as part of the segments interpretation below.*
d. ***sequenced data****: the Analyzer breaks down the audio into musically relevant elements that occur sequenced in time.*

Dari data yang kecil ke data yang besar di sisipkan :

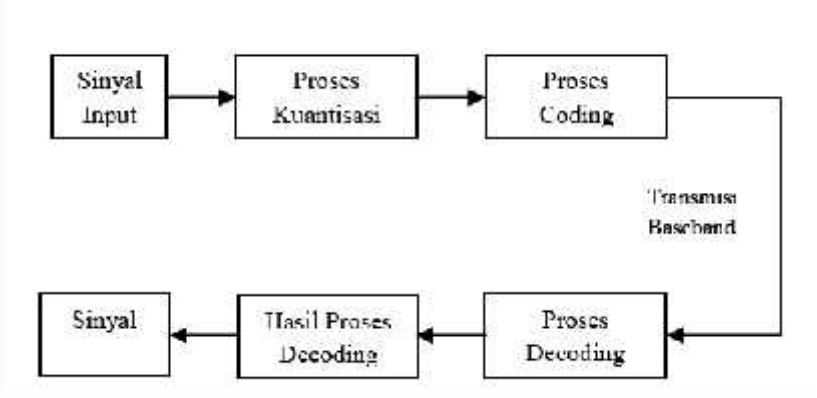
a. ***segments****: a set of sound entities (typically under a second) each relatively uniform in timbre and harmony. :*
    1. *loudness_start: indicates the loudness level at the start of the segment*
    2. *loudness_max_time: offset within the segment of the point of maximum loudness*
    3. *loudness_max: peak loudness value within the segment*
b. ***tatums****: list of tatum markers, in seconds.*
c. ***beats****: list of beat markers, in seconds.*
d. ***bars****: list of bar markers, in seconds.*
e. ***sections****: a set of section markers, in seconds.*

Informasi yang didapatkan pada saat menggunggah *file* MP3 adalah berupa teks dengan informasi ID lagu (“song_id”: SOZIABG12A81C21DD2), ID *upload* (“id”: TRFNINZ1331CF52919), judul lagu (“title”: When You‟re Gone), nama artis (“artist”: Avril Lavigne),dan lainnya. Dari ID *upload* diatas digunakan kembali pada *url link* untuk mendapatkan fitur-fitur dari *file* MP3 yang diunggah. *URL* yang digunakan adalah sebagai berikut :

http://developer.echonest.com/api/v4/track/profile?api_key=HGUBAV9BYLZP7GQ9P&format=xml&id=TRYAQMX1335AD3097C&bucket=audio_summary

Output dari *link* diatas :

Penggunggahan *file* MP3 menghasilkan informasi berupa judul lagu, artis, id lagu, energi, mode, *key*, *loudness*, dan tempo. Website *Echonest* mengekstrak file *audio* yang diunggah dengan melakukan analisis terhadap sinyal *audio* dari *file* tersebut. Salah satu metode yang dilakukan pihak Echonest untuk melakukan analisis signal adalah dengan metode *Pulse-Code Modulation* (PCM). PCM adalah proses perubahan dari sinyal analog menjadi sinyal digital serta bilangan biner, dengan melalui bermacam-macam proses mulai dari *sampling*, kuantisasi, *coding* setelah itu ditransmisikan dengan transmisi *baseband* lalu diproses dengan *decoding*, menghasilkan sinyal hasil *decoding* dan menghasilkan *output* sinyal seperti sinyal hasil sampling. Berikut skema dari gambaran sistem PCM dapat dilihat pada gambar 2.5.



Gambar 2.5 Pemrosesan Sinyal dalam PCM

(Sumber : Astawa, 2012)

2.4 Sinyal

Diambil dari berbagai sumber, pengertian sinyal sangat bermacam, antara lain :

a. Fungsi satu variabel atau lebih yang menunjukkan informasi dalam fisik fenomena alam.
b. Sistem berupa arus data yang mengalir melalui jalur transmisi.
c. Suatu indikator yang digunakan sebagai alat komunikasi.
d. Suatu impuls atau fluktuasi besaran listrik seperti tegangan, arus, kuat medan listrik, yang mengkodekan informasi.
e. Suatu impuls elektronik atau gelombang radio yang dikirim atau diterima.
f. Suatu kuantitas/besaran yang berubah-ubah.
g. Tegangan listrik yang dihasilkan oleh mikrofon sebagai respon terhadap ucapan ‘should’ dan ‘we’. Tegangan tersebut bersesuaian dengan tekanan akustik pada telinga, yang merupakan reaksi terhadap perubahan tekanan.

Dalam proses pengolahan sinyal analog, sinyal input masuk ke Analog Signal Processing (ASP), diberi berbagai perlakukan (misalnya pemfilteran, penguatan, dsb.) dan outputnya berupa sinyal analog. Proses pengolahan sinyal secara digital memiliki bentuk sedikit berbeda. Komponen utama system ini berupa sebuah processor digital yang mampu bekerja apabila inputnya berupa

sinyal digital. Untuk sebuah input berupa sinyal analog perlu proses awal yang bernama digitalisasi melalui perangkat yang bernama *analog-to-digital conversion* (ADC), dimana sinyal analog harus melalui proses sampling, *quantizing dan coding*. Demikian juga output dari processor digital harus melalui perangkat *digital-to-analog conversion* (DAC) agar outputnya kembali menjadi bentuk analog. Ini bisa kita amati pada perangkat seperti PC, digital sound system, dsb.

2.4.1 Klasifikasi Sinyal

Klasifikasi sinyal dapat dibeda-bedakan menurut :

A. Sinyal analog dan sinyal digital

Sinyal analog adalah sinyal yang mempunyai nilai untuk setiap waktu, sinyal ini bersifat kontinyu terhadap waktu. Sinyal digital adalah sinyal yang tidak untuk setiap waktu terdefinisi, sinyal ini bersifat diskrit terhadap waktu. Sinyal digital berasal dari sinyal analog yang disampling, yang artinya mengambil nilai suatu sinyal analog mulai t=0, t=Δt, t=2Δt, t=3Δt dan seterusnya. Untuk mendapatkan sinyal waktu diskrit yang mampu mewakili sifat sinyal aslinya, proses sampling harus memenuhi syarat Nyquist :

$$fs > 2\ fi \qquad (2.2)$$

Keterangan :

f_s = frekuensi sinyal sampling

f_i = frekuensi sinyal informasi yang akan disampel

B. Sinyal riil dan sinyal kompleks

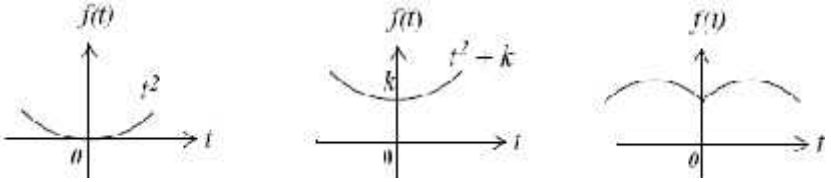
Sinyal riil merupakan sinyal yang bersifat riil untuk semua variabel. Sinyal kompleks merupakan sinyal yang mempunyai nilai yang kompleks ada faktor nilai imajiner.

C. Sinyal ganjil dan genap

Sinyal genap mempunyai sifat:

- f(-t) = f (t)
- Polinomial dengan pangkat yang genap

  cost = $1-\frac{t^2}{2!}+\frac{t^4}{4!}-\frac{t^6}{6!}$ ……
- Contoh :



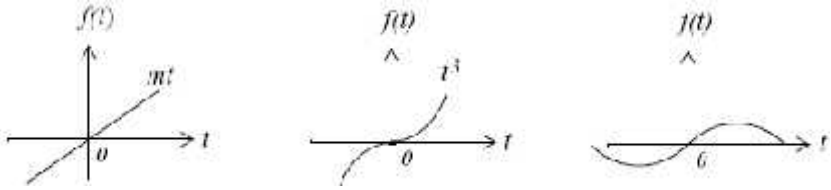
Gambar 2.6 Sinyal Genap

(Sumber : Jehan, 2012)

Sinyal ganjil mempunyai sifat *:*

- –f(-t) = f (t)
- Polinomial dengan pangkat ganjil

  $\sin t = t-\frac{t^3}{3}+\frac{t^5}{5}-\frac{t^7}{7}$ ……
- Contoh :

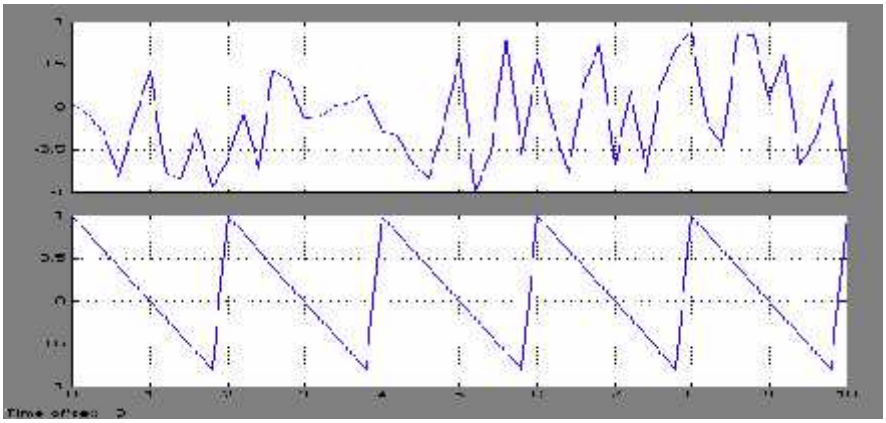


Gambar 2.7 Sinyal Ganjil

(Sumber : Jehan, 2012)

D. Sinyal deterministik dan sinyal random

Sinyal deterministik merupakan sinyal yang nilainya secara lengkap untuk semua titik waktu sudah dikenal. Sinyal random mempunyai nilai random untuk waktu yang diberikan. Nilai-nilai sinyal random untuk setiap titiknya tidak diketahui dengan pasti, sehingga sinyal random hanya dibahas berdasarkan karakter statistik, misalnya nilai rata-rata dan nilai tengah.



Gambar 2.8 Sinyal Deterministik dan sinyal Random

(Sumber : Astawa, 2012)

2.5 *Application Programming Interface (API)*

Antarmuka pemrograman aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi.

2.6 K-Nearest Neighbors

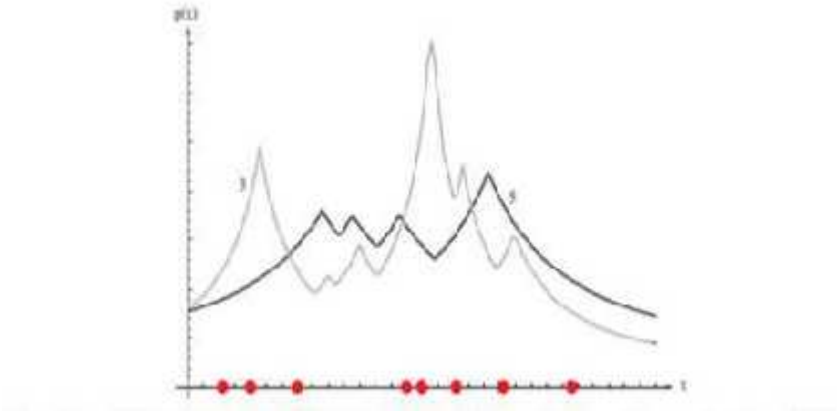
K-Nearest Neighbor (KNN) adalah suatu metode yang menggunakan algoritma supervised dimana hasil dari query instance yang baru diklasifikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru bedasarkan atribut dan training sample. Classifier tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik query, akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik query. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek. algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari query instance yang baru.

Algoritma metode KNN sangatlah sederhana, bekerja berdasarkan jarak terpendek dari query instance ke training sample untuk menentukan KNN-nya. Training sample diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi training sample. Sebuah titik pada ruang ini

ditandai kelac c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut.

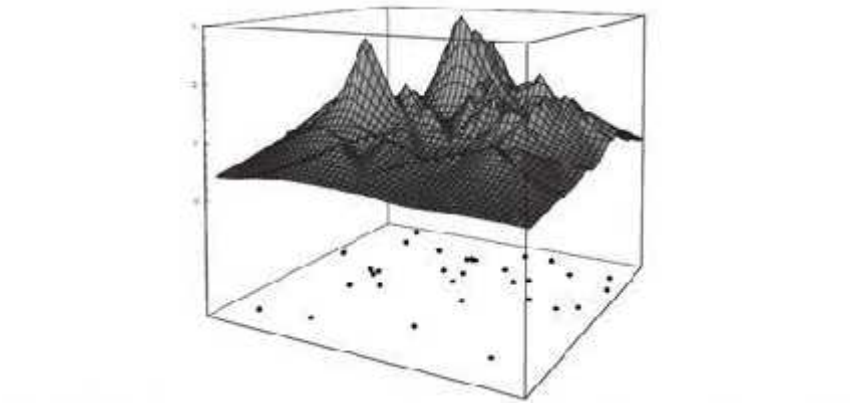
Pada fase training, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data (yang klasifikasinya tidak diketahui). Jarak dari vektor baru yang ini terhadap seluruh vektor training sample dihitung dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Sebagai contoh, untuk mengestimasi p(x) dari n training sample dapat memusatkan pada sebuah sel disekitar x dan membiarkannya tumbuh hingga meliputi k samples. Samples tersebut adalah KNN dari x. Jika densitasnya tinggi di dekat x, maka sel akan berukuran relatif kecil yang berarti memiliki resolusi yang baik. Jika densitas rendah, sel akan tumbuh lebih besar, tetapi akan berhenti setelah memasuki wilayah yang memiliki densitas tinggi.



Gambar 2.9 Delapan titik dalam satu dimensi dan estimasi densitas KNN dengan k = 3 dan N = 5

(Sumber : Astawa, 2012)



Gambar 2.10 KNN mengistimasi densitas dua dimensi dengan k = 5
(Sumber : Astawa, 2012)

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan training data yang paling dekat (dengan kata lain, k = 1) disebut algoritma nearest neighbor.

Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap training data yang memiliki banyak noise dan efektif apabila training data-nya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat), training berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap query instance pada keseluruhan training sample. Konsep dasar algoritma KNN adalah mencari jarak terdekat antara data yang dievaluasi dengan K-tetangga (*neighbors*) terdekatnya dalam data uji. Secara umum algoritma KNN bisa dilihat sebagai berikut :

1. Menentukan nilai parameter K (jumlah tetangga terdekat).
2. Menghitung jarak setiap sampel data dengan data yang akan diuji.
3. Mengurutkan data berdasarkan jarak dari yang terkecil hingga yang terbesar.
4. Klasifikasi data baru dari hasil perhitungan dengan memberikan label kelas tiap data.

Jarak untuk setiap data uji dengan data sampel dihitung dengan rumus jarak *Euclidean* seperti pada persamaan.

$$\text{Dist } (x,y) = \sqrt{\sum_{x=y}^{D}(x_j - y_j)^2} \quad (2.3)$$

Keterangan :

dist (x,y) adalah jarak euclidean antara vektor *x* dan vektor *y* ;

x adalah komponen ke *j* dari vektor *x* ;

y adalah komponen ke *j* dari vektor *y* ;

D adalah jumlah komponen pada vektor *i* dan vektor *k*.

Beberapa kelebihan dari algoritma KNN ini adalah sebagai berikut :

1. Handal dalam uji data yang memiliki tingkat *noise* yang tinggi.
2. Efektif jika data yang diuji dalam jumlah besar.
3. Algoritma secara umum mudah untuk dimengerti.

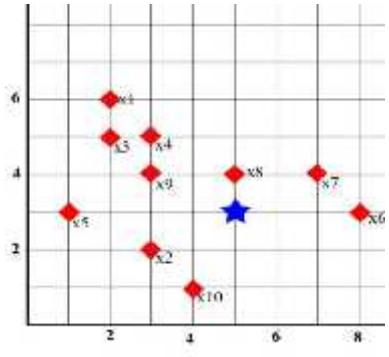
Contoh persoalan dari *K-Nearest Neighbour*

Terdapat 10 buah titik X dengan koordinat kartesius dengan nilai :

- X1(2,6)
- X2(3,2)
- X3(2,5)
- X4(3,5)
- X5(1,3)
- X6(8,3)
- X7(7,4)
- X8(5,4)
- X9(3,4)
- X10(4,1)

Diberikan suatu titik uji dengan nilai Y(5,3) dengan nilai K = 5.

Hitunglah jarak *Euclidean*nya !



Gambar 2.11 Gambar koordinat titik uji dan titik sampel

Untuk jarak *Euclidean* setiap titik X terhadap Y dapat dihitung sebagai berikut :

(X,Y) $= \sqrt{(xi-yi)^2+(xj-yj)^2}$

(X1,Y) = $\sqrt{(2-5)^2+(6-3)^2} = 2\sqrt{3}$

(X2,Y) = $\sqrt{(3-5)^2+(2-3)^2} = \sqrt{5}$

(X3,Y) = $\sqrt{(2-5)^2+(5-3)^2} = 2\sqrt{2}$

(X4,Y) = $\sqrt{(3-5)^2+(5-3)^2} = 2\sqrt{2}$

(X5,Y) = $\sqrt{(1-5)^2+(3-3)^2} = 4$

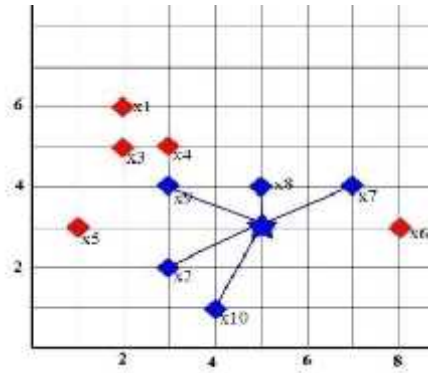
(X6,Y) = $\sqrt{(8-5)^2+(3-3)^2} = 3$

(X7,Y) = $\sqrt{(7-5)^2+(4-3)^2} = \sqrt{5}$

(X8,Y) = $\sqrt{(5-5)^2+(4-3)^2} = 1$

(X9,Y) = $\sqrt{(3-5)^2+(4-3)^2} = \sqrt{5}$

(X10,Y) = $\sqrt{(4-5)^2+(1-3)^2} = \sqrt{5}$



Gambar 2.12 Solusi Permasalahan KNN

Gambar 2.14 diatas menunjukkan bahwa *Nearest Neighbors* dari titik uji adalah titik X8, X2, X7, X9, dan X10 karena memiliki jarak terdekat dari titik uji. Dalam penelitian ini akan digunakan *Cosine Similarity* untuk mengurangi kemungkinan adanya nilai kemiripan yang sama antar beberapa data pada jarak *Euclidean Distance*. Karena dalam *Cosine Similarity* selain memperhitungkan jarak pada data uji dan data sampel, sudut yang dibentuk antar 2 data tersebut juga diperhitungkan dalam penentuan data yang termasuk dalam *Nearest Neighbors*.

Saat nilai *similarity* ditemukan, maka akan ditentukan data yang merupakan *Nearest Neighbours* dari data uji sebanyak nilai *K*. Nilai K yang dimaksud adalah banyaknya tetangga terdekat atau data yang memiliki kemiripan tertinggi di antara semua data yang ada.

Nilai yang digunakan dalam acuan sistem ini adalah nilai *threshold*, nilai *threshold* merupakan nilai batas yang suatu parameter kerja yang ditentukan untuk mengetahui kinerja suatu sistem. Unjuk kerja suatu sistem dikatakan baik jika parameter-parameter kinerja sistem tidak melebihi nilai *threshold* yang ditentukan.

2.7 *Cosine Similarity*

Cosine Similarity merupakan salah satu metode *similarity* yang memperhitungkan jarak dan sudut antar data uji dengan data sampel. Menurut Steinbach didalam penelitian suta, “Perhitungan pada *cosine similarity* didasarkan pada *vector dot product* dan jarak dari data uji ke data sampel.” Untuk menentukan nilai *similarity* dalam *cosine similarity* dapat dicari dengan persamaan.

$$\text{Sim (A, B)} = \text{cosine } \theta = \frac{X.Y}{|X||Y|} = \frac{x1'x2 \cdot y1'y2}{(x1^2+y1^2)^{\frac{1}{2}} \quad (x2^2+y2^2)^{\frac{1}{2}}} \qquad (2.4)$$

Keterangan :

x dan y adalah value untuk A

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2 \sum_{i=1}^{n}(y_i-\bar{y})^2} \quad (2.5)$$

Keterangan :

r adalah nilai koefisien korelasi, berkisar antara -1.0 sampai +1.0

x dan y adalah value untuk A dan B.

Contoh persoalan dari *Cosine Similarity*

Tentukan *Cosine Similarity* jika diketahui data uji adalah X(2,5) dengan data sampel adalah

.Y1(2,6) .Y2(3,2) .Y3(2,1) .Y4(3,5) .Y5(1,1)

Dari soal diatas dapat diselesaikan dengan rumus :

$$\text{Sim (X,Y)} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2 \sum_{i=1}^{n}(y_i-\bar{y})^2}$$

$$\text{Sim (X, Y1)} = \frac{(2*2+5*6)}{\sqrt{2*2+5*5}\ \sqrt{(2*2+6*6)}} = \frac{34}{\sqrt{29}\sqrt{40}} = \frac{34}{5.38*6.32} = 0.999$$

$$\text{Sim (X, Y2)} = \frac{(2*3+5*2)}{\sqrt{2*2+5*5}\ \sqrt{(3*3+2*2)}} = \frac{16}{\sqrt{29}\sqrt{13}} = \frac{16}{5.38*3.6} = 0.826$$

$$\text{Sim (X, Y3)} = \frac{(2*2+5*1)}{\sqrt{2*2+5*5}\ \sqrt{(2*2+1*1)}} = \frac{9}{\sqrt{29}\sqrt{5}} = \frac{9}{5.38*2.23} = 0.75$$

$$\text{Sim (X, Y4)} = \frac{(2*3+5*5)}{\sqrt{2*2+5*5}\ \sqrt{(3*3+5*5)}} = \frac{31}{\sqrt{29}\sqrt{34}} = \frac{31}{5.38*5.83} = 0.988$$

$$\text{Sim (X, Y5)} = \frac{(2*1+5*1)}{\sqrt{2*2+5*5}\ \sqrt{(1*1+1*1)}} = \frac{7}{\sqrt{29}\sqrt{2}} = \frac{7}{5.38*1.41} = 0.922$$

Dari hasil perhitungan nilai *cosine similarity* di atas, dapat dilihat bahwa data1 (2,6) merupakan data yang memiliki kemiripan tertinggi dengan data uji X(2,5) dengan nilai *cosine similarity* 0.999.

Contoh Penerapan KNN dan CS dalam pencarian musik :

Tabel 2.2 Informasi data dari echonest

DATA						
Artis	**Judul**	**keys**	**mode**	**loudness**	**energi**	**tempo**
Rihanna	Disturbia	4	1	-5.738	0.621906	124.999
Adele	Someone like u	9	1	-10.725	0.182099	135.219

Queen	I Want To Break Free	4	1	-18.058	0.400647	108.717
Madonna	Sorry	10	0	-9.506	0.677908	135.985
Christina	Beautiful	5	0	-12.03	0.618485	74.048

Data Query :						
Artis	**judul**	**keys**	**mode**	**loudness**	**energi**	**tempo**
Track 01	?	7	1	-9.908	0.534867	110.067

Tabel 2.3 Hasil Gabungan Metode KNN dan Cosine Similarity

KNN	**Cosine-Similarity**			
(X,Y)	**X.Y (i)**	**||X||**	**||Y(i)||**	**(X,Y)**
15.79117158	13844.44967	110.7393292	125.2000855	0.998548544
25.2470798	15053.51037	110.7393292	135.9457125	0.999931805
100.4631846	12174.287	110.7393292	110.2843505	0.996845281
26.11368968	15132.00903	110.7393292	136.6848339	0.999710777
36.1507709	8304.765263	110.7393292	75.18782965	0.99741966

2.8 *Min Max Normalization*

Perbedaan rentang data yang ada pada data sampel, bisa menyebabkan proses identifikasi memiliki akurasi yang lebih rendah. Karena bisa menyebabkan satu dimensi data tidak mempengaruhi nilai *distance* atau *similarity* pada suatu identifikasi. Untuk itulah diperlukan suatu normalisasi data untuk menyetarakan rentang nilai antar dimensi pada data yang akan identifikasi. *Min Max Normalization* (Matlab,2009) didapatkan dari perhitungan nilai minimum dan maksimum dari suatu fitur data, atau dapat dilihat pada persamaan.

$$f' = \frac{f - fmin}{fmax - fmin} \tag{2.6}$$

Keterangan :

f’ = nilai fitur yang telah dinormalisasi dari suatu fitur lagu

f = nilai fitur sebelum dinormalisasi

f min = rentang nilai terkecil untuk *f*

f max = rentang nilai terbesar untuk *f*

2.9 *Confusion Matrix*

Evaluasi dari sistem yang akan dibangun sangat penting dilakukan guna mengetahui kualitas dari sistem yang dibangun. Tingkat akurasi sistem sangat berpengaruh pada kualitas yang dihasilkan. Akurasi sistem dihitung setelah tahap pengkodean dilakukan. Analisis dilakukan terhadap tingkat keberhasilan dari algoritma *K-Nearest Neighbor* yang digunakan pada penelitian ini. Akurasi sistem dihitung dengan metode *confusion matrix* (Kohavi,1998). Dimana nilai yang dicari adalah nilai akurasi. Dimana untuk menghitung akurasi digunakan rumus :

$$\text{Acc} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (2.7)$$

Serta akan dicari nilai *precision* dan *recall* dari penelitian ini, dengan rumus :

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (2.8)$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (2.9)$$

Keterangan :

TP (*True Positive*) = Jumlah sampel positif yang berhasil diklasifikasikan

TN(*True Negative*) = Jumlah sampel negatif yang berhasil diklasifikasikan

FP (*False Positive*) = Jumlah sampel positif yang tidak terklasifikasi

FN(*False Negative*) = Jumlah sampel negatif yang tidak terklasifikasi