BAB II

TINJAUAN PUSTAKA

2.1 Profil Instansi

Museum Geologi atau Museum Geologi Bandung merupakan salah satu objek wisata sejarah dan edukasi di kota kembang Bandung. Museum ini memberikan informasi dan pengetahuan mengenai geologi atau ilmu bumi dengan cara yang menarik dan menyenangkan. Di museum ini, wisatawan dapat menyaksikan berbagai material geologi berbentuk batuan, fosil, ataupun mineral yang ditemukan di berbagai wilayah Indonesia. Karena pentingnya peranan museum dan koleksi yang dimiliki, Museum Geologi Bandung tercatat sebagai warisan nasional dan dilindungi dalam Peraturan Pemerintah. [1].

2.1.1 Sejarah Instansi

Keberadaan Museum Geologi sangat erat kaitannya dengan sejarah penyelidikan geologi di Indonesia yang telah dimulai sejak 1850-an. Pada saat itu, lembaga yang mengkoordinasikan penyelidikkan geologi adalah *Dienst van het Mijnwezen*. Museum Geologi untuk pertama kalinya diresmikan pada saat pembukaan gedung *Dienst van den Mijnbouw* yaitu pada 16 Mei 1929. Peresmian ini bertepatan dengan pembukaan kongres Ilmu Pengetahuan Pasifik ke-IV yang diselenggarakan di Institut Teknologi Bandung. Gedung ini berfungsi sebagai perkantoran yang dilengkapi dengan sarana laboratorium geologi dan museum untuk menyimpan dan memperagakan hasil survei geologi.

Sejalan dengan dinamika sejarah, secara kelembagaan Museum Geologi terus mengalami perubahan. Pada zaman pemerintahan Belanda (1929-1941), Museum Geologi disebut *Geologisch Laboratorium* dan merupakan unit kerja dari *Dienst van het Mijnwezen* yang berganti nama menjadi *Dienst van den Mijnbouw*. Kemudian pada zaman pendudukan Jepang (1942-1945). *Dienst van den Mijnbouw* diganti namanya menjadi *Kogyoo Zimusho* yang kemudian berganti nama menjadi *Tisitutyosazyo* dimana Museum Geologi sebagai bagian dari Laboratorium Paleontologi dan Kimia.

Pengelolaan Museum Geologi yang semula berada dibawah Pusat Djawatan Tambang dan Geologi (PDTG), berganti nama menjadi : Djawatan Pertambangan Republik Indonesia (1950-1952), Djawatan Geologi (1952-1956), Pusat Djawatan Geologi (1956-1957), Djawatan Geologi (1957-1963). Pada tahun 1963-1978 berganti nama menjadi Direktorat Geologi, setelah itu berganti nama menjadi Pusat Penelitian dan Pengembangan Geologi (1978-2002). Pada 2003 Museum Geologi menjadi Unit Pelaksana Teknis Museum Geologi (UPT MG), di bawah Pusat Survei Geologi, Badan Geologi, Kementerian Energi dan Sumberdaya Mineral. Pada 2013, berdasarkan Permen ESDM No. 12 Tahun 2013, Museum Geologi menjadi Unit Pelaksana Teknis Museum Geologi (UPT MG), di bawah Badan Geologi, Kementerian Energi dan Sumber daya Mineral [1].

2.1.2 Logo Instansi

Berikut adalah logo dari instansi tempat penulis melakukan penelitian yaitu logo Museum Geologi Bandung seperti pada gambar 2.1.



Gambar 2.1 Logo Museum Geologi [1]

Museum Geologi memiliki logo seperti yang digambarkan diatas. Logo tersebut menganalogikan tentang keberadaan dan fungsi Museum Geologi yang erat kaitannya dengan segala aspek geologis. Dibawah ini adalah makna dari logo di atas:

1. Gambar lingkaran dengan garis vertikal dan horizontal adalah bumi
2. Segitiga yang berada dia atas adalah udara
3. Segitiga yang di bawah adalah menggambarkan tanah
4. Segitiga yang ada di kanan adalah api
5. Segitiga yang di kiri adalah air

2.1.3 Visi dan Misi

Museum geologi merupakan Unit pelaksana Teknis di lingkungan Badan Geologi. Adapun visi dan misi yang terdapat pada Museum Geologi Bandung [9].

A. Visi

Menjadi sumber Informasi Geologi Indonesia

B. Misi

1. Memperagakan dan Mengkomunikasikan koleksi Museum
2. Menyediakan Informasi dan materi edukasi geologi
3. Mendokumentasikan dan mengkonsevasikan koleksi museum
4. Melakukan penelitian koleksi dan pengembangan museum
5. Melakukan pameran museum dan geologi
6. Melakukan penyuluhan dan sosialisasi geologi
7. Melakukan kerjasama dengan intasi dan sekolah
8. Melakukan pengelolaan museum secara professional
9. Memberikan pelayanan jasa permuseuman

2.1.4 Badan Hukum Instansi

Museum menurut *International Council of Museums* (ICOM) adalah sebuah lembaga yang bersifat tetap, tidak mencari keuntungan, melayani masyarakat dan perkembangannya, terbuka untuk umum, memperoleh, merewat, menghubungkan, dan memamerkan artefak-artefak perihal jati diri manusia dan lingkungannya untuk tujuantujuan studi, pendidikan dan rekreasi. Sedangkan Museum menurut Peraturan Pemerintah No. 19 Tahun 1995 Pasal 1 ayat (1) adalah lembaga, tempat penyimpanan, perawatan, pengamanan, dan pemanfaatan benda-benda bukti materiil hasil budaya manusia serta alam dan lingkungannya guna menunjang upaya perlindungan dan pelestarian kekayaan budaya bangsa [1].

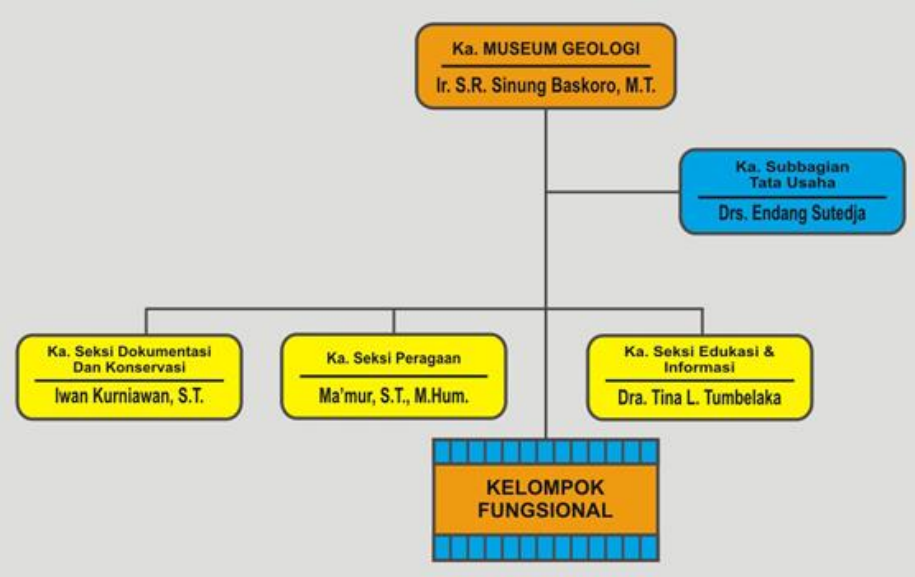
Pendirian sebuah museum memiliki acuan hukum, yaitu:

1. Undang-undang RI Nomor 5 Tahun 1992 tentang Benda Cagar Budaya.
2. Peraturan Pemerintah Nomor 10 Tahun 1993 tentang Pelaksanaan Undang-undang RI Nomor 5 Tahun 1992.

3. Peraturan Pemerintah Nomor 19 Tahun 1995 tentang Pemeliharaan dan Pemanfaatan Benda Cagar Budaya di Museum.
4. Keputusan Menteri Kebudayaan dan Pariwisata Nomor KM.33/PL.303/MKP/2004 tentang Museum.
5. Peraturan Menteri Energi dan Sumber Daya Mineral (PerMen ESDM) No. 12 tahun 2013 tentang Organisasi dan Tata Kerja Museum Geologi.

2.1.5 Sturktur Organisasi dan *Job Description*

Dalam Peraturan Menteri Energi dan Sumber Daya Mineral (PerMen ESDM) No. 12 tahun 2013 tentang Organisasi dan Tata Kerja Museum Geologi, dengan struktur organisasi seperti pada gambar 2.2 [10].



Gambar 2.2 Struktur Organisasi Museum Geologi [10]

A. Tugas

Melaksanakan pengelolaan, penelitian, pengembangan, konservasi, peragaan dan penyebarluasan informasi koleksi geologi.

B. Fungsi

Dalam melaksanakan tugasnya, Museum Geologi menyelenggarakan fungsi:

1. Penyusunan rencana dan program serta pengelolaan kerja sama.
2. Pelaksanaan pengelolaan koleksi geologi
3. Pelaksanaan penelitian, pengembangan dan konservasi koleksi geologi.

4. Pelaksanaan peeragaan dan pameran koleksi geologi.
5. Pelaksanaan bimbingan edukasi dan penyebarluasan informasi koleksi geologi.
6. Pengelolaan sarana dan prasarana
7. Pelaksanaan ketatausahaan, kepegawaian, keuangan dan rumah tangga.

C. Job Description

1. Tugas Kepala Museum Geologi

   Tugas Pokok:

   Kepala Museum Geologi mempunyai tugas teknis penunjang dan operasional untuk melaksanakan penelitian, pengembangan, dan konservasi serta memperagakan koleksi geologi, kemudian dalam melaksanakan tugasnya [10].

   Fungsi :

   a. Penyiapan rencana dan program penelitian, pengembangan, konservas, peragaan, dan publikasi koleksi Geologi.
   b. Pelaksanaan pengelolaan dan pengembangan dokumentasi.
   c. Pelaksanaan pengelolaan dan pengembangan dan peragaan.
   d. Pelaksanaan penelitian dan pengembangan serta publikasi.
   e. Pelaksanaan dan pengembangan kerjasama serta pelayanan jasa permuseuman.
   f. Pelaksanaan ketatausahaan, kepegawaian, keuangan, dan rumah tangga.
   g. Evaluasi pelaksanaan rencana dan program penelitian, pengembangan, konservasi, peragaan, dan publikasi koleksi geologi.

2. Kelompok Kerja Subbagian Tata Usaha

   Tugas Pokok:

   Melaksanakan penyiapan bahan penyusunan program dan laporan, urusan ketatausahaan, kepegawaian, keuangan, serta rumah tangga.

3. Kelompok Kerja Seksi Dokumentasi

   Tugas Pokok:

   Museum Geologi mempunyai peran untuk mendokumentasikan koleksi geologi yang terdiri dari batuan, mineral, dan fosil, termasuk dokumen lainnya yang sangat penting bagi sejarah dan perkembangan ilmu geologi di masa yang akan datang. Koleksi batuan, mineral, dan fosil ini juga merupakan data yang dikonversikan sehingga menjadi koleksi yang abadi untuk generasi yang akan datang, tetapi juga dapat menunjang kegiatan eksplorasi, baik sumber daya mineral maupun energi di Indonesia karena koleksi tersebut merupakan data geologi dari seluruh wilayah Indonesia. Pendokumentasian koleksi batuan, mineral, dan fosil tersebut menjadi tugas Seksi Dokumentasi, maka diperlukan pembersihan secara khusus terlebih dahulu disamping pembuatan preparat untuk penelitian koleksi tersebut. Dari hasil penelitian, maka informasi tersebut disimpan di ruang dokumentasi dimana segala informasi mengenai koleksi tersebut disimpan sebagai *database* [10].

4. Kelompok Kerja Seksi Peragaan

   Tugas Pokok:

   Seksi peragaan harus mampu memelihara peragaan yang telah ada juga sebaiknya dapat melakukan pengembagan.

5. Kelompok Kerja Seksi Edukasi dan Informasi

   Tugas Pokok:

   Seksi edukasi dan informasi harus mampu dapat menyampaikan informasi geologi kepada pengunjung sesuai dengan tingkat pendidikannya.

2.2 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dengan pengembangan aplikasi diantaranya mengenai pengertian teknologi *multitouch*, *library* TUIO, Flash, perangkat lunak *Reactivision*, *communication protocol*, pemodelan UML dan hal-hal yang berguna dalam proses analisis permasalahan serta *tools* pendukung lainnya.

2.2.1 Batu Artefak

Artefak atau *artifact* merupakan benda arkeologi atau peningalan benda-benda bersejarah, yaitu semua benda yang dibuat atau dimodifikasi oleh manusia yang dapat dipindahkan. Contoh artefak adalah alat-alat batu, logam dan tulang, gerabah, prasasti lempeng dan kertas, senjata-senjata logam (anak panah, mata panah, dll), *terracotta* dan tanduk binatang. Artefak dalam arkeologi mengandung pengertian benda atau bahan alam yang jelas dibuat oleh (tangan) manusia Ciri penting dalam konsep artefak adalah bahwa benda ini dapat bergerak atau dapat dipindahkan (*movable*) oleh tangan manusia dengan mudah (*relatif*) tanpa merusak atau menghancurkan bentuknya. Artefak adalah sisa-sisa alat bekas suatu kebudayaan zaman *prehistory* yang digali dari dalam lapisan bumi. Artefak ialah objek yang dibentuk atau diubah oleh manusia [2].

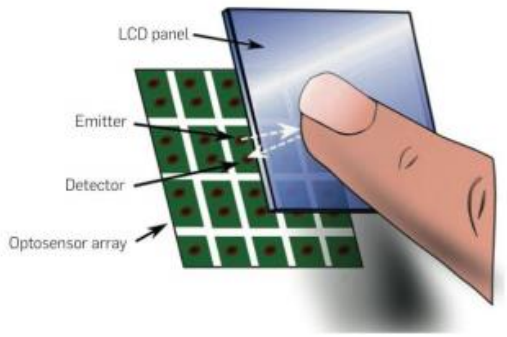
2.2.2 Multitouch

Touch screen adalah sebuah perangkat input komputer yang bekerja dengan adanya sentuhan pada layar menggunakan jari atau pena digital *(stylus)*. Pada tahun 1971, pertama kali "*Touch* Sensor" ini dikembangkan oleh Doktor Sam Hurst (pendiri Elographics) sekaligus sebagai seorang instruktur di *University of Kentucky*. Sensor ini disebut "Elograph," dan telah dipatenkan oleh *University of Kentucky Research Foundation*. "Elograph" ini tidak transparan seperti *touchscreens* modern, namun demikian elograph telah menjadi tonggak sejarah yang signifikan dalam teknologi *touchscreen*. Pada tahun 1974 *touchscreen* pertama sesunggunya yang telah dilengkapi dengan permukaan transparan dikembangkan oleh Doktor Sam Hurst dan Elographics. Pada tahun 1977 Elographics dikembangkan dan dipatenkan dengan teknologi lima-kawat resistif.

Dr Andrew Hsu, seorang ahli teknologi layar sentuh, menyatakan bahwa "layar IR (inframerah) adalah permukaan yang paling tahan lama dan dapat menangani lingkungan yang sulit, membuat mereka cocok untuk aplikasi militer ". Meskipun penelitian ini tidak akan berfokus pada teknologi dalam hal aplikasi militer, Dapat dilihat bahwa layar sentuh dengan teknologi inframerah, akan menjadi permukaan paling tahan lama, dan juga paling serbaguna [13].

Dalam teknologi ini, LED inframerah (*light emitting dioda*) yang disusun di sisi berlawanan diatas permuakaan kaca. Dioda memproyeksikan cahaya inframerah ke sensor yang terletak di seberang sensor. Sensor membaca kekuatan pancaran sinar inframerah, dan "ketika pengguna membuat kontak dengan layar, sistem mengukur penurunan sinyal *output* sensor, pengukuran ini memungkinkan sistem untuk menghitung lokasi sentuhan". Hal ini menunjukan bahwa ketika jari menyentuh layar, pancaran cahaya inframerah terhalang oleh jari pengguna, namun sebagian pancaran cahaya bisa lolos dan melewatkan cahaya terhadap sensor. Sensor mengirim pengukuran cahaya ke sistem, dimana sistem akan menganalisis data dan mengetahui di mana pengguna menyentuh. Teknologi ini sudah memiliki kemampuan multi-touch karena pancaran cahaya inframerah tidak pernah sepenuhnya terhalang oleh sentuhan pengguna.

Tipe kedua dari sistem inframerah membutuhkan lebih banyak tempat daripada yang pertama. Sistem ini didasarkan pada refleksi; pancaran cahaya yang dipancarkan dari dalam kaca dan sebagian cahaya keluar melalui lensa sedangkan sebagian lain masuk kembali ke unit. Kamera ditempatkan di dalam kaca dan dikalibrasi dengan standar refleksi sehingga, Ketika objek seperti jari menyentuh permukaan, memancarkan cahaya pada titik kontak, menyebabkan pantulan di dalam akrilik maka kamera dibawah permukaan menangkap hasil penyebaran dan mengirimkan informasi ke perangkat lunak pengolah gambar, yang dapat membaca beberapa sentuhan secara simultan dan menerjemahkannya ke dalam perintah. Dalam contoh lain, terhadap layar kaca, memungkinkan untuk menggunakan Lensa tipis yang hanya tahan lama, lebih murah, dan lebih tangguh seperti gambar 2.3.



Gambar 2.3 *Transmitter* dan *receiver* Inframerah layar sentuh tipis

1. *Microsoft Surface*

Pengaplikasian refleksi layar sentuh infra merah adalah *Microsoft Surface*. Di awal tahun 2001, "Steve Bathiche dari *Microsoft Hardware* dan Andy Wilson dari *Microsoft Research*", memulai pengembangan *surface*. Proses berlanjut pada pembangunan sampai dengan tahun 2003 ketika ide ini disampaikan kepada Bill Gates, dan 85 prototipe diciptakan untuk pengembang pada *Surface Computing Group*, sebuah kelompok yang diciptakan pada tahun 2004 hanya untuk mengembangkan *Surface*. Pada tahun 2005, berbagai prototipe yang dibuat dan dianalisis dan akhirnya, pada tahun 2008, *Microsoft Surface* dijual kepada publik dengan harga $ 12.500 atau $ 15.000.

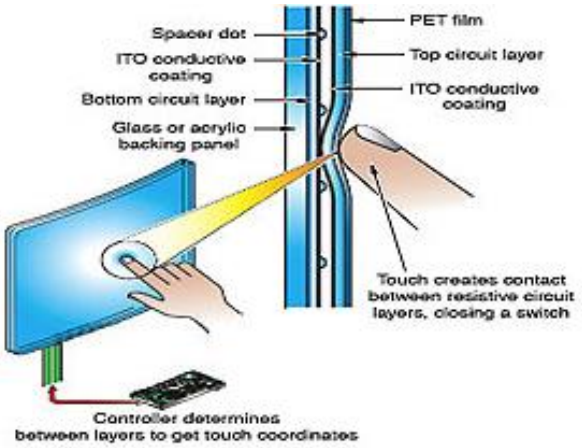
Teknologi *surface* ini menggunakan beberapa sensor gambar sekitar satu sisi dari permukaan sentuh dan lampu inframerah pada sisi lain. Ketika pengguna menempatkan jarinya di permukaan, menghalangi sinar inframerah, maka perangkat akan memproyeksikan bayangan. Menggunakan beberapa kamera, unit mengkonversi bayangan ini menjadi titik sentuh melalui triangulation". Seperti yang dinyatakan oleh Dr Hsu, *Surface* menggunakan sistem kamera dan LED Inframerah untuk menangkap gambar dari objek yang menyentuh permukaan. Teknologi *Surface* mengambil langkah maju dengan teknologi mereka, mengembangkan *motherboard* khusus untuk menghitung data yang ditransmisikan pada 100 *megabyte* per detik, dari kamera. Perangkat lunak ini dapat mendeteksi dan membedakan berbagai jenis sentuhan. *Surface* mengkategorikan sentuhan menjadi tiga set: *finger*, *blobs* dan *object*, seperti gambar 2.4.



Gambar 2.4 Microsoft Surface

2. *Resistive Touch Screen*

Sistem layar sentuh resistif adalah jenis yang paling umum dari teknologi layar sentuh di pasar saat ini. Perangkat ini digunakan dalam berbagai aplikasi, seperti ponsel, genggam permainan, perangkat navigasi GPS, dan bahkan beberapa digital kamera. Teknologi layar sentuh resistif beroperasi dengan cara yang sangat sederhana. Layar ini dibangun menggunakan dua lapis bahan konduktif Indium Tin Oxide (ITO), dipisahkan oleh celah udara kecil. Lapisan bawah umumnya adalah kaca dan bagian atas adalah bahan fleksibel, atau seringkali plastik. Ketika pengguna menekan ke bawah pada lapisan ITO atas, fisik membungkuk untuk melakukan kontak dengan bagian bawah lapisan ITO, mengubah perlawanan dari dua lapisan. Tipikal layar sentuh resistif menggunakan 4 kabel, 2 kabel pada setiap panel. Seperti yang terlihat pada Gambar 3, masing-masing panel sesuai dengan yang berbeda sumbu. Sumbu tegak lurus ini memungkinkan komputer untuk mengambil pengukuran perubahan tahanan dari masing-masing panel, dan menghitung posisi titik sentuh dari yang X dan komponen Y, seperti gambar 2.5.

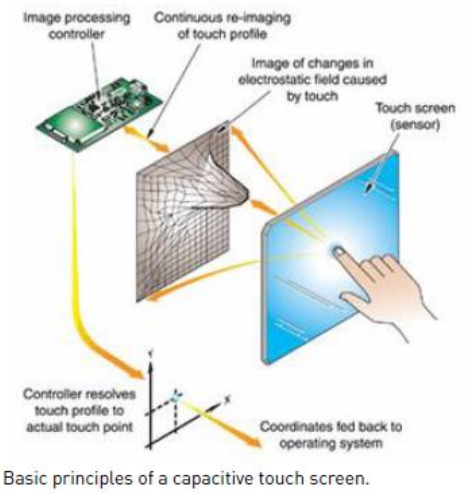


Gambar 2.5 Layar Sentuh Resistif

3. *Capacitive Touch Screen*

Sangat *smooth* untuk dioperasikan karena hanya membutuhkan keberadaan jari tangan, tanpa tekanan. Tampilan layarnya memiliki kejernihan hingga sekitar 90% dan bila terkena paparan sinar matahari tidak mempengaruhi kualitas tampilan. Namun kelemahannya adalah tidak mendukung penggunaan

stylus. Terlebih lagi jenis layar sentuh yang ini tidak bisa digunakan sama sekali saat menggunakan sarung tangan, seperti gambar 2.6.



Gambar 2.6 Layar Sentuh *Capacitive*

Teknologi *multitouch* adalah fitur penting dari layar sentuh *capacitive* yang mempunyai kemampuan untuk mengenali dan menghitung beberapa titik sentuhan pada satu waktu. Teknologi *multitouch* telah ada sejak penelitian awal di University of Toronto pada tahun 1982. *multitouch* sebenarnya adalah nama dagang yang diterapkan oleh Apple karena sifatnya yang mendeteksi banyak sentuhan.

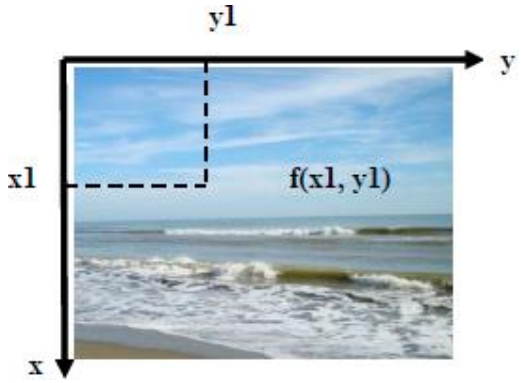
Multitouch telah diterapkan dengan berbagai cara berbeda, tergantung pada ukuran dan jenis antarmuka. Prinsip kerja suatu meja multitouch memproyeksikan melalui kaca akrilik atau kaca dan lampu latar belakang dengan LED. Ketika sebuah jari atau benda menyentuh permukaan menyebabkan cahaya terpecah, refleksi tertangkap dengan sensor atau lampu kamera yang mengirim data ke perangkat lunak yang merespon terhadap sentuhan.

Perangkat *multitouch* dapat berbeda tergantung dari bahan dan alat yang digunakan, bahan utama yang umum digunakan adalah infrared dan kamera optik. Poin penting yang mempengaruhi hasil sentuhan adalah rata-rata penangkapan gambar per detik pada kamera dan kualitas kamera.

2.2.3 Citra Digital

Citra digital adalah citra dua dimensi yang dapat ditampilkan pada layar monitor komputer sebagai himpunan berhingga (diskrit) nilai digital yang disebut *pixel* (*picture elements*). *Pixel* adalah elemen citra yang memiliki nilai yang menunjukkan intensitas warna. Berdasarkan cara penyimpanan atau pembentukannya, citra digital dapat dibagi menjadi dua jenis. Jenis pertama adalah citra digital yang dibentuk oleh kumpulan *pixel* dalam *array* dua dimensi. Citra jenis ini disebut citra *bitmap* atau citra *raster*. Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik *vektor*. Citra digital (diskrit) dihasilkan dari citra analog (*kontinu*) melalui digitalisasi. Digitalisasi citra analog terdiri s*ampling* dan *quantitazion. Sampling* adalah pembagian citra ke dalam elemen-elemen diskrit (*pixel*), sedangkan *quantitazion* adalah pemberian nilai intensitas warna pada setiap *pixel* dengan nilai yang berupa bilangan bulat [14].

Citra digital dapat didefinisikan sebagai fungsi dua *variabel*, *f(x,y)*, dimana *x* dan *y* adalah kordinat spasial dan nilai *f(x,y)* adalah *brightness level* dari citra pada kordinat tersebut, seperti gambar 2.7.



Gambar 2.7 Citra Digital

2.2.4 Pengolahan Digital

Operasi pengolahan citra digital umumnya dilakukan dengan tujuan memperbaiki kualitas suatu gambar sehingga dapat dengan mudah diinterpretasikan oleh mata manusia dan untuk mengolah informasi yang ada pada suatu gambar untuk kebutuhan identifikasi objek secara otomatis. Operasi-operasi yang dilakukan

di dalam pengolahan citra banyak jenisnya. Secara umum, operasi pengolahan citra dapat diklasifikasikan dalam beberapa [15].

1. *Image enhancement*.

   Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi *parameter* citra. Dengan operasi ini, ciri-ciri khusus yang terdapat di dalam citra lebih ditonjolkan. Contoh-contoh operasi perbaikan citra:

   a. Perbaikan kontras gelap/terang (*Image brightness*)

      Pengubahan kecerahan gambar dilakukan untuk membuat citra lebih terang atau lebih gelap. Kecerahan gambar dapat diperbaiki dengan menambahkan (atau mengurangkan) sebuah konstanta kepada (atau dari) setiap piksel di dalam citra.

   b. Peregangan kontras (*contrast stretching*)

      Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) di dalam sebuah gambar. Citra dapat dikelompokkan ke dalam tiga kategori kontras : citra kontras rendah (*low contrast*), citra kontras bagus (*good contrast/normal contrast*), dan citra kontras tinggi (*high contrast*).

   c. Pengubahan *histogram* citra – Perataan histogram

      Nilai-nilai intensitas di dalam citra diubah sehingga penyebarannya seragam (*uniform*). Tujuan dari perataan *histogram* adalah untuk memperoleh penyebaran *histogram* yang merata, sedemikian sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama. *grayscaling*

   d. Pelembutan Citra (*Image smoothing*)

      Pelembutan citra bertujuan untuk menekan gangguan (*noise*) pada citra.

   e. Penajaman (*sharpening*)

      Operasi penajaman citra bertujuan memperjelas obyek di dalam citra dengan menghilangkan bagian citra yang lembut.

   f. Pengubahan geometrik

      Koreksi geometrik dilakukan pada citra yang memiliki gangguan yang terjadi pada waktu proses perekaman citra, misalnya pergeseran kordinat citra (*translasi*), perubahan ukuran citra, dan perubahan orientasi kordinat

citra (*skew*). Koreksi geometri yang sederhana adalah dengan operasi geometri sederhana, seperti rotasi, translasi, dan penskalaan citra.

g. Pewarnaan semu (*pseudocoloring*)

Pewarnaan semu adalah proses memberi warna tertentu pada nilai-nilai piksel suatu citra berdasarkan kriteria tertentu, misalnya suatu warna tertentu untuk suatu interval derajat keabuan tertentu.

2. *Image Restoration*.

Operasi ini bertujuan menghilangkan atau meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra. Bedanya, pada pemugaran citra penyebab degradasi gambar diketahui. Contoh-contoh operasi pemugaran citra:

a. penghilangan kesamaran (*deblurring*)
b. penghilangan *noise*

3. *Image Compression*.

Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus. Contoh metode pemampatan citra adalah metode JPEG.

4. *Image Segmentation*.

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengenalan pola.

5. *Image Analysis*.

Jenis operasi ini bertujuan menghitung besaran kuantitif dari citra untuk menghasilkan deskripsinya. Teknik analisis citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek dari sekelilingnya. Contoh operasi analisis citra:

a. Pendeteksian tepi objek (*edge detection*)
b. Ekstraksi batas (*boundary*)
c. Representasi daerah (*region*)

6. *Image Reconstruction*

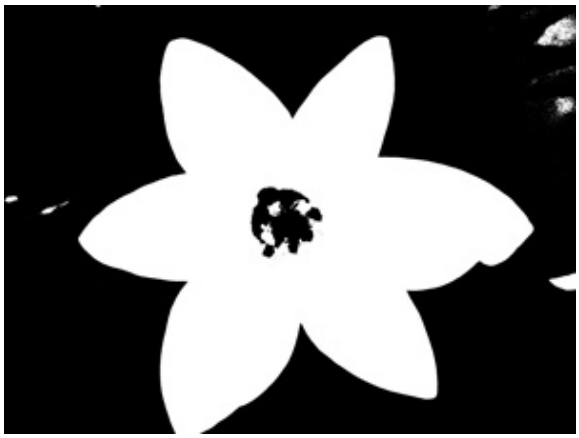
Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Misalnya beberapa foto *rontgen* dengan sinar *X* digunakan untuk membentuk ulang gambar organ tubuh.

7. *Image Matching*

Pencocokan citra merupakan usaha untuk mengidentifikasi dan mengukur derajat kesamaan atau kecocokan objek pada dua atau lebih citra.

2.2.5 Citra Biner

Citra biner merupakan citra yang telah melalui proses pemisahan piksel-piksel berdasarkan derajat keabuan yang dimiliki. Citra biner adalah citra yang hanya direpresentasikan nilai tiap pikselnya dalam satu bit (satu nilai *binary*). Banyaknya warna yang terdapat pada citra biner adalah dua, yaitu hitam dan putih [16]. Salah satu contoh dari gambar biner dapat dilihat pada Gambar 2.8. Dibutuhkan satu *bit* di *memori* untuk menyimpan kedua warna ini. Setiap piksel pada citra bernilai 0 untuk hitam dan 1 untuk putih [17].



Gambar 2.8 Contoh Citra Biner

2.2.6 Citra Warna

Citra warna adalah citra dengan system grafik yang memiliki satu set niai tersusun *(a set of ordered values)* yang menyatakan berbagai tingkat warna. Citra warna bukanlah seperti citra *grayscale.* Dimana setiap set nilai tersusun mewakili

satu *'scale'* warna atau *'hue'*. Sistem yang dipakai untuk mewakili warna yaitu sistem *RGB (Red, Green, Blue).* Sistem *RGB* adalah sistem penggabungan antara warna-warna primer *(additive primary colours)* yaitu merah *(Red)*, Hijau *(Green)* dan Biru *(Blue)* untuk memperoleh warna tertentu. Misalnya warna putih diperoleh dari hasil gabungan warna merah = 255, hijau = 255 dan biru = 255. Dalam sistem *RGB*, warna putih cerah dinyatakan dengan *RGB* (255, 255,255). *Range* nilai dari setiap warna *primer* adalah 0 sampai 255. Sehingga kemungkinan warna yang dapat terbentuk dengan sistem *RGB* adalah 256 x 256 x 256 yakni kurang lebih 16.7 juta warna. Pada table dibawah ini diperlihatkan beberapa kode warna hasil gabungan warna *RGB* [18]. Di bawah ini merupakan contoh tabel warna untuk sistem warna *RGB* atau *Red, Green Blue*, seperti pada tabel 2.1.

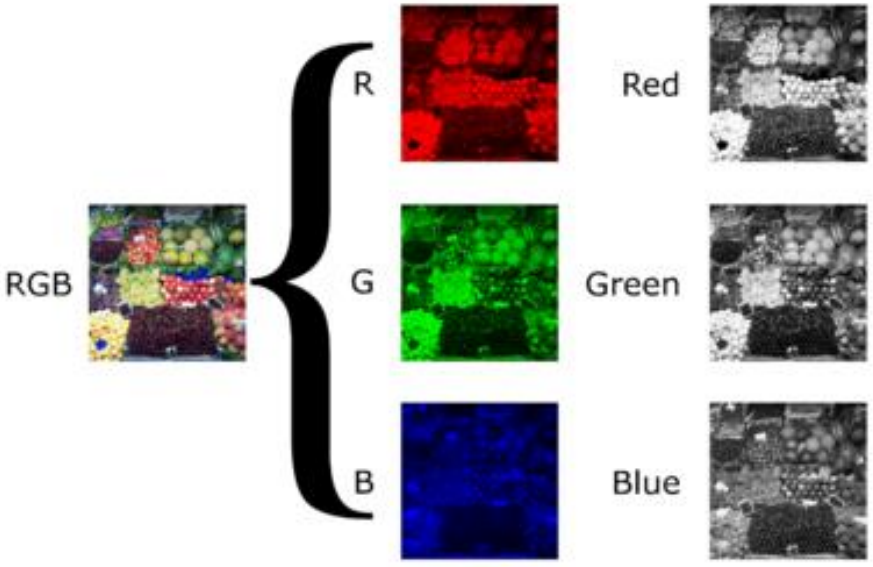
Tabel 2.1 Kode Warna RGB

Colour	***Red***	***Green***	***Blue***
Black	0	0	0
Red	255	0	0
Blue	0	0	255
Green	0	255	0
Cygan	0	255	255
Magenta	255	0	255
Yellow	255	255	0
White	0	0	0
Gray	128	128	128

2.2.7 Grayscaling

Proses *grayscaling* merupakan proses konversi gambar dari gambar berwarna (RGB) menjadi gambar satu warna *(monochrome)*. Pada mode *grayscaling* memanfaatkan warna gray (abu-abu) sebanyak 256 tingkat gradasi dan disimpan dalam format 8 bit untuk setiap *sample pixel*. Format ini sangat membantu dalam pemrograman karena manipulasi *bit* yang tidak terlalu banyak. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra *grayscale* dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan [19].

Setiap *pixel* dari gambar *grayscale* mempunyai nilai *brightness* (kecerahan) antara 0 (hitam) hingga 255 (putih). Nilai *grayscale* dapat juga diartikan seperti suatu tinta yang berwarna hitam, dimana mempunyai tingkat kehitaman yang bervariasi (0% sama dengan putih, sedangkan 100% adalah hitam). Warna abu-abu merupakan satu-satunya warna pada ruang RGB dengan komponen merah, hijau, dan biru mempunyai intensitas yang sama, seperti pada gambar 2.9.



Gambar 2.9 Contoh *Grayscaling*

2.2.8 Image Processing

Image Processing adalah manipulasi dan analisis suatu informasi gambar. Gambar yang dimaksud disini adalah gambar visual dalam dua dimensi. Konsep dasar dari sistem *image processing* diambil dari kemampuan indera penglihatan manusia yang selanjutnya dihubungkan dengan kemampuan otak manusia. *Image Processing* dalam sejarahnya telah diaplikasikan dalam berbagai cabang ilmu. *Image Processing* menyangkut pula berbagai gabungan cabang-cabang ilmu, seperti diantaranya optik, elektronik, matematika, fotografi dan teknologi komputer. Dalam bidang ilmu komputer, *image processing* merupakan salah satu bagian pemrosesan data berupa gambar, baik gambar foto maupun gambar bergerak (video) [19].

2.2.9 Noise

Pada saat proses *capture* (pengambilan gambar), ada beberapa gangguan yang mungkin terjadi, seperti kamera tidak fokus atau munculnya bintik-bintik yang bisa jadi disebabkan oleh proses *capture* yang tidak sempurna. Setiap gangguan pada citra dinamakan dengan *noise*. *Noise* pada citra tidak hanya terjadi karena ketidak-sempurnaan dalam proses *capture*, tetapi bisa juga disebabkan oleh kotoran-kotoran yang terjadi pada citra. Berdasarkan bentuk dan karakteristiknya. *Noise* pada citra dibedakan menjadi beberapa macam yaitu:

(1) *Gaussian*

(2) *Speckle*

(3) *Salt & Pepper*

Macam-macam noise ini dapat dilihat pada gambar 2.10 berikut ini:



(a) (b) (c)

Gambar 2.10 *noise* (a) *gaussian* (b) *speckle* dan (c) *salt* & *pepper*

Noise speckle merupakan model *noise* yang memberikan warna hitam pada titik yang terkena *noise*. Sedangkan *noise salt & pepper* seperti halnya taburan garam, akan memberikan warna putih pada titik yang terkena *noise*. Pada beberapa pengolahan citra, terkadang untuk menguji suatu algoritma untuk dapat mereduksi *noise*, maka *noise* dihasilkan dari proses pembangkitan *noise*. Untuk membangkitkan *noise* digunakan suatu bilangan acak sebagai pengganti *noise* yang dihasilkan.

2.2.10 Noise Gaussian

Noise gaussian merupakan model *noise* yang mengikuti distribusi normal standar dengan rata-rata nol dan standard deviasi 1. Efek dari *gaussian noise* ini, pada gambar muncul titik-titik berwarna yang jumlahnya sama dengan prosentase

noise. *Noise* gaussian dapat dibangkitkan dengan cara membangkitkan bilangan acak [0,1] dengan distribusi *gaussian*. Kemudian untuk piksel yang terkena *noise*, nilai fungsi citra ditambahkan dengan nilai *noise* yang ada, atau dirumuskan dengan:

$$y(i,j) = x(i,j) + p.a$$

dimana: a = nilai bilangan acak berdistribusi *gaussian*

p = prosentase *noise*

y(i,j) = nilai citra terkena *noise*.

x(i,j) = nilai citra sebelum terkena *noise*.

Faktor elektronik seperti kurang pencahayaan, suhu yang terlalu tinggi menyebabkan timbulnya *noise Gaussian*. Untuk membangkitkan bilangan acak berdistribusi *gaussian*, tidak dapat langsung menggunakan fungsi *random*, tetapi diperlukan suatu metode yang digunakan untuk mengubah distribusi bilangan acak ke dalam fungsi *f* tertentu [20].

g = imnoise(f,'gaussian',m,var)

Dimana nilai *default* untuk m = 0 dan var = 0.01

2.2.11 Filter Gaussian

Pada *filter Gaussian*, nilai intensitas setiap piksel diganti dengan rata-rata dari nilai pembobotan untuk setiap piksel-piksel tetangganya dan piksel itu sendiri. Filter harus dirancang terlebih dahulu, dengan berdasarkan pada ordo matriks dan nilai standar deviasi σ^2. Dengan efek kurva yang dihasilkan, maka akan didapat efek *smoothing* pada citra yang diproses, Semakin besar nilai standart deviasi, maka semakin halus pula efek yang dihasilkan dari pemfilteran menggunakan Filter yang dihasilkan [20].

Persamaan fungsi *zero mean Gaussian* dua variabel Untuk mengisi elemen-elemen pada matriks kernel *Gaussian.*

$$g(x,y) = e^{-(x^2+y^2)/(2.\ \sigma^2)}$$

Persamaaan distribusi diskrit gaussian

$$g(x,y) = c.\ e^{-(x^2+y^2)/(2.\ \sigma^2)}$$

Dimana nilai c yang dihasilkan dikalikan dengan masing-masing bobot nilai, sehingga menghasilkan matriks *filter gaussian*. *e* Adalah konstanta *euler* dengan nilai 2.718281828.

Filter gaussian ini sebenarnya hampir sama dengan filter rata-rata hanya ada nilai bobot yang tidak rata seperti pada filter rata-rata, tetapi mengikuti fungsi *Gaussian* untuk mereduksi *noise* sebagai berikut:

$$G(x,y) = \frac{1}{s\sqrt{\pi}} e^{-\left((x-m_x)^2+(y-m_y)^2\right)/2s}$$

dimana : **s** adalah sebaran dari fungsi *gaussian*

(m_x,m_y) adalah titik tengah dari fungsi *gaussian*

2.2.12 Background Subtraction

Definisi dari *background subtraction* adalah proses untuk menghilangkan *background* dari suatu citra. *Background subtraction* ini dilakukan melalui pengurangan setiap *frame* masukan dari kamera (*grayscale*) dengan suatu citra background (*grayscale*) dengan ukuran citra yang sama. Proses pengurangan tersebut akan menghasilkan citra yang berisi perubahan nilai keabuan, sehingga menandakan terdapat piksel-piksel (komponen citra) yang berbeda dari *background* atau terjadi perubahan nilai piksel pada *frame* tersebut. Persamaaan pengurangan *frame* (citra *grayscale*) dapat di tulis sebagai berikut :

$$\boldsymbol{S_{(n)} = |F_{(n)} - B|}$$

Dimana : $\boldsymbol{S_{(n)}}$ = Matrik hasil perubahan nilai keabuan ke-*n*

$\boldsymbol{F_{(n)}}$ = Matrik *frame* ke –*n*

$\boldsymbol{B}$ = Matrik citra *background*

S diperoleh dengan menghitung perbedaan antara semua pasangan piksel pada kordinat yang bersesuaian dengan ***F*** dan ***B***. Teknik ini dapat digunakan untuk latar belakang yang kompleks dengan syarat kamera yang dipakai harus tetap diam dan tidak terjadi perubahan pencahayaannya selama proses berjalan. Artinya, jika terjadi perubahan posisi kamera atau perubahan intensitas cahaya, proses

inisialisasi latar belakang harus diulangi kembali. Kelemahan lainnya adalah, teknik ini tidak dapat mendeteksi objek yang berwarna sama dengan latar belakang [15].

2.2.13 Histogram

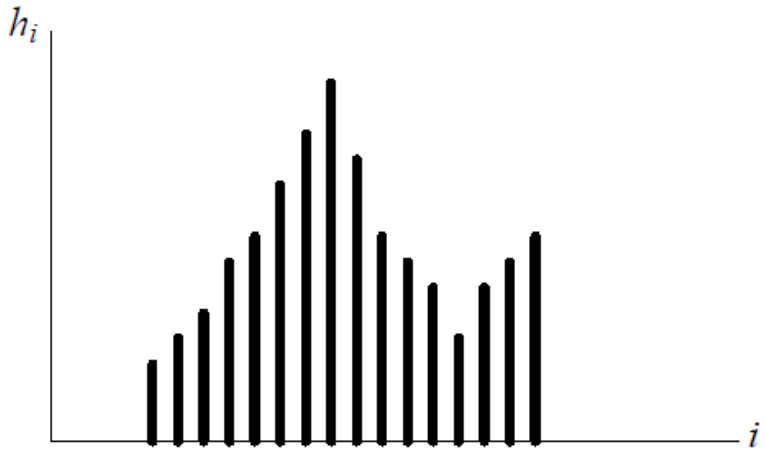
Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah *histogram* dapat diketahui frekuensi kemunculan *noise* (*relative*) dari intensitas pada citra tersebut. *Histogram* juga dapat menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontas (*contrast*) dari sebuah gambar. Karena itu, *histogram* adalah alat bantu yang berharga dalam pekerjaan pengolahan citra baik secara kualitatif maupun kuantitatif. Secara matematis *histogram* citra dihitung dengan rumus :

$$h_i = \frac{n_i}{n} \quad , i = 0, 1, \ldots, L-1$$

Dimana n_i = jumlah *pixel* yang memiliki derajat keabuan *i*

n = jumlah seluruh *pixel* di dalam citra

Secara grafis *histogram* ditampilkan dengan diagram batang. Perhatikan dari persamaan diatas bahwa nilai n_i telah dinormalkan dengan membaginya dengan *n*. Nilai h_i berada di dalam selang 0 sampai 1, gambar 2.11 merupakan contoh *histogram* [21].



Gambar 2.11 Contoh *Histogram*

2.2.14 Metode Thresholding

Thresholding citra adalah suatu metode yang digunakan untuk memisahkan antara obyek dan *background*. *Thresholding* adalah teknik yang sederhana tapi efektif untuk segmentasi citra. Proses *Thresholding* sering disebut dengan proses binerisasi. Dalam proses *thresholding* terhadap sebuah citra, hasil yang diperoleh tidak selalu memuaskan dan sesuai dengan keinginan. Hal ini dikarenakan faktor penghambat seperti pencahayaan yang tidak merata atau citra yang kabur yang menyebabkan *histogram* tidak bisa dipartisi dengan baik.

Terdapat berbagai metode dalam memilih *Threshold*. Metode paling sederhana dilakukan dengan cara memilih nilai *mean* atau *median*. Pada dasanya jika piksel objek lebih terang dibandingkan dengan *background* maka piksel objek tersebut juga lebih terang dari rata-ratanya. Pada gambar yang masih memiliki *noise* dengan *background* dan nilai objek, *mean* dan *median* akan bekerja maksimal dalam *threshold*. Metode *thresholding* secara umum dibagi menjadi dua, yaitu :

1. *Thresholding* global

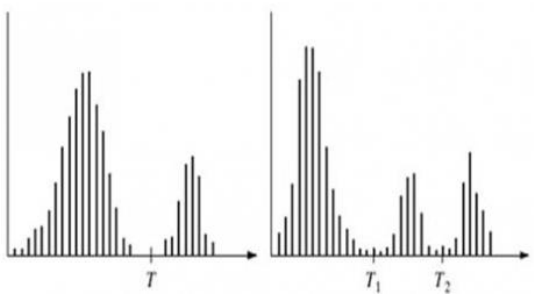
   Thresholding dilakukan dengan mempartisi histogram dengan menggunakan sebuah *threshold* (batas ambang) global T, yang berlaku untuk seluruh bagian pada citra.
2. *Thresholding* adaptif

   Thesholding dilakukan dengan membagi citra menggunakan beberapa sub citra. Lalu pada setiap sub citra, segmentasi dilakukan dengan menggunakan *threshold* yang berbeda.

Persamaan untuk menentukan nilai *thresholding* dapat dirumuskan sebagai berikut :

$$T = \frac{f_{maks} + f_{min}}{2}$$

Thresholding dikatakan global jika nilai *threshold* T hanya bergantung pada *f(x,y)*, yang melambangkan tingkat keabuan pada titik *(x,y)* dalam suatu citra. Berikut ini akan disajikan contoh partisi *histogram* untuk memperoleh *threshold* dalam Gambar 2.12 [18].



Gambar 2.12 Contoh Partisi Histogram Untuk Memperoleh Nilai Threshold

Histogram yang berada pada sisi kiri Gambar 1 mewakili citra *f(x,y)* yang tersusun atas obyek terang di atas *background* gelap. Piksel-piksel obyek dan *background* dikelompokkan menjadi dua *mode* yang dominan. Cara untuk mengekstraks obyek dari *background* adalah dengan memilih nilai *threshold* T yang memisahkan dua *mode* tersebut. Kemudian untuk sembarang titik *(x,y)* yang memenuhi *f(x,y) > T* disebut titik obyek, selain itu disebut titik *background*. Kesuksesan metode ini bergantung pada seberapa bagus teknik partisi *histogram*. Citra hasil *thresholding* dapat didefinisikan sebagaimana persamaan di bawah ini.

$$g(x,y) = \begin{cases} 1 & if \quad f(x,y) > T \\ 0 & if \quad f(x,y) \le T \end{cases}$$

2.2.15 Tracking

Penjejakan Objek Bergerak(*Obyek Tracking*)adalah proses mencari lokasi dari objek yang akan diamati dari suatu data video untuk setiap satu satuan *frame* dalam data video tersebut. Metode yang sering digunakan dalam penjejakan objek bergerak berdasarkan pengenalan objek diantaranya :

1. *SIFT*
2. *RANSAC*
3. *Mean Shift*
4. *Camshift*

Sementara metode yang sering digunakan dalam penjejakan objek bergerak berdasarkan gerakan objek diantaranya :

1. *Kalman filter*
2. *Extended Kalman filter*
3. *Particle filter*

2.2.16 Metode Kalman Filter

R.E Kalman pertama kali memiliki ide tentang KF pada tahun 1958. Kemudian pada tahun 1960, R.E Kalman mempublikasikan papernya yang berisi suatu penyelesaian rekursif pada masalah *filtering* yang didasarkan pada penggunaan teknik ruang keadaan dan algoritma rekursif untuk mengestimasi keadaan sistem dinamik yang kemudian dikenal dengan metode KF [22].

Dalam proses estimasi, secara umum KF memiliki dua tahapan yaitu tahap prediksi (*time update*) dan tahap koreksi (*measurement update*). Tahap prediksi dipengaruhi oleh dinamika sistem, sedangkan tingkat akurasinya dihitung menggunakan persamaan kovariansi *error*. Pada tahap koreksi, hasil estimasi *variable* keadaan yang diperoleh pada tahap prediksi, dikoreksi menggunakan data pengukuran. Salah satu bagian dari tahap ini yaitu menentukan matriks *Kalman Gain* yang digunakan untuk meminimumkan kovariansi *error*. Kedua tahapan ini diulang sampai waktu *k* yang ditentukan.

KF sederhana hanya dapat diaplikasikan pada sistem linier. Tetapi pada dunia nyata sistem nonlinier ada baik pada model sistem atau model pengukuran atau keduanya. Oleh karena itu dikembangkanlah metode EKF yang dapat digunakan untuk mengestimasi model sistem nonlinier dengan kondisi model sistem dan model pengukuran menjadi fungsi turunan dari variabel keadaan. Fungsi turunan dari variabel keadaan membentuk matriks Jacobi [23]. Sama halnya dengan KF, estimasi EKF dilakukan dengan dua tahapan yaitu dengan cara memprediksi variabel keadaan berdasar sistem dinamik yang disebut tahap prediksi (*time update*) dan tahap koreksi (*measurement update*) terhadap data-data pengukuran untuk memperbaiki hasil estimasi.

KF pada hakikatnya tidak dapat langsung diterapkan untuk sistem *non-linear* dan *non-Gaussian* (distribusi kompleks dan multimoda). Untuk itu EKF melakukan *linearisasi* dari sistem *non-linear* tersebut agar bisa diestimasi oleh KF. Dengan demikian, EKF dapat melakukan pelacakan menggunakan model *non-linier* dan *non-Gaussian*.

2.2.17 Image Matching

Pencocokan citra merupakan usaha untuk mengidentifikasi dan mengukur derajat kesamaan atau kecocokan objek pada dua atau lebih citra. Metode pencocokan citra (*image matching*) digunakan dalam berbagai aplikasi pengolahan citra untuk keperluan otomatisasi proses dan menentukan seberapa mirip/sama bentuk objek baik secara semantik maupun geometrik antara citra yang satu dengan citra yang lainnya. Pada penelitian ini untuk proses pencocokan citra menggunakan metode *graph matching* dalam pengukuran kesamaan citra, *transformasi* (perubahan) citra menjadi bentuk *graph* merupakan hal utama. Sekumpulan piksel yang membentuk *region* dalam *graph* disebut *verteks*. Region yang saling berdekatan atau bertetangga dalam *graph* disebut sebagai *region adjacency graph* akan membentuk sebuah garis hubung antar *verteks* yang disebut *edge* [2].

2.2.18 Graph Matching

Graph matching adalah salah satu metode yang digunakan dalam pencocokan citra. Metode *graph matching* dapat memudahkan dan mempercepat pengenalan objek pada teknik pencocokan citra. Dari proses tersebut didapat hasil bahwa penentuan posisi atau titik koordinat pengambilan *fragmen* (*crop*) dari citra masukan akan mempengaruhi nilai *edit* operasi dasar. Semakin besar nilai *threshold* yang digunakan maka nilai jarak kesamaan citra yang dihasilkan semakin kecil. Pada citra yang di-*resize* dan tidak di-*resize* dengan threshold 0 menghasilkan jarak kesamaan citra minimum yaitu 1 di titik koordinat *crop* sehingga semua citra pembanding dikatakan sama dengan citra acuan. Pemberian threshold 0 pada pengujian citra yang di-*resize*, tidak di-*resize*, dan diberi filter menghasilkan jarak kesamaan citra yang nilainya besar. Sekumpulan piksel yang membentuk *region* dalam *graph* disebut verteks. *Region* yang saling berdekatan atau bertetangga dalam *graph* disebut sebagai *region adjacency graph* akan membentuk sebuah garis hubung antar verteks yang disebut *edge*.

Graph adalah salah satu pokok bahasan matematika diskrit yang telah lama dikenal dan banyak diaplikasikan pada berbagai bidang. Secara umum, *graph* G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi G = (V,

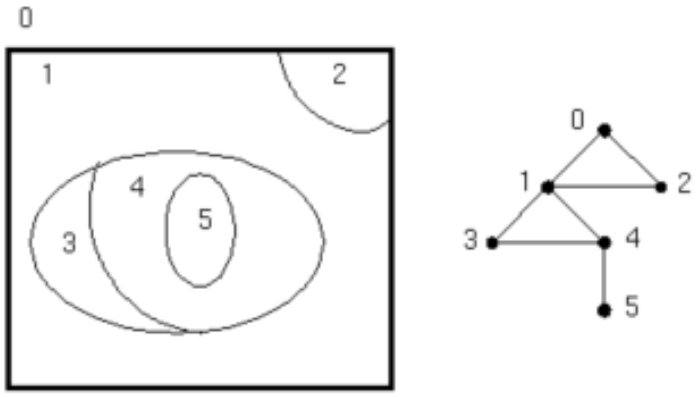
E) yang dalam hal ini V adalah tidak kosong dari simpul-simpul (*nodes*) dan E adalah himpunan sisi atau busur (*edges*) yang menghubungkan sepasang simpul [24].

Berdasarkan orientasi arah pada sisi, maka secara umum *graph* dibedakan atas dua jenis, yaitu :

1. Graf tak-berarah (*undirected graf*) adalah Graf yang sisinya tidak mempunyai orientasi arah.
2. Graf berarah (*directed graf* atau *digraf*) adalah Graf yang setiap sisinya mempunyai orientasi arah.

2.2.19 Region Adjacency Graph

Region adjacency graph adalah salah satu nilai yang dihubungkan dengan *nodes* dan *arc* (*edge*) sebagai nilai baru dari terbentuknya sebuah daerah (wilayah), yang mana harus dimasukan dalam algoritma yang diberikan yaitu sebagai nilai *nodes* yang tersambung pada dua *region* R_i dan R_j [25].



Gambar 2.13 Contoh *Region Adjacency Graph*

Pada gambar 2.13 diatas tiap wilayah atau area direpresentasikan dengan simpul, sedangkan sisi menyatakan hubungan ketetanggaan antar area. Dalam pengolahan citra, sebuah gambar dapat disegmentasi menjadi beberapa wilayah sesuai karakteristik warna dan batasan (*boundary*) lalu dibuat graf pemetaannya yang menggambarkan ketetanggaan antar area. Graf ini biasa disebut juga *Region Adjacency Graph*. Dari sinilah kemudian graf tersebut dicocokkan dengan bentuk yang ada dalam *database* sehingga bentuk dari objek pada gambar dapat dikenali.

2.2.20 Adobe Flash

Adobe Flash (dahulu bernama Macromedia Flash) adalah salah satu perangkat lunak komputer yang merupakan produk unggulan Adobe *Systems*. Adobe Flash digunakan untuk membuat gambar vektor maupun animasi gambar tersebut. Berkas yang dihasilkan dari perangkat lunak ini mempunyai *file extension* .swf dan dapat diputar di penjelajah web yang telah dipasangi Adobe Flash *Player*. Flash menggunakan bahasa pemrograman bernama *ActionScript* yang muncul pertama kalinya pada Flash 5. Sebelum tahun 2005, Flash dirilis oleh Macromedia. Flash 1.0 diluncurkan pada tahun 1996 setelah Macromedia membeli program animasi vektor bernama *FutureSplash*. Versi terakhir yang diluncurkan di pasaran dengan menggunakan nama 'Macromedia' adalah Macromedia Flash 8. Pada tanggal 3 Desember 2005 Adobe *Systems* mengakuisisi Macromedia dan seluruh produknya, sehingga nama Macromedia Flash berubah menjadi Adobe Flash, seperti gambar 2.14.



Gambar 2. 14 Adobe Flash CS6

Adobe Flash merupakan sebuah program yang didesain khusus oleh Adobe dan program aplikasi standar *authoring tool professional* yang digunakan untuk membuat animasi dan bitmap yang sangat menarik untuk keperluan pembangunan situs web yang interaktif dan dinamis. Flash didesain dengan kemampuan untuk membuat animasi 2 dimensi yang handal dan ringan sehingga flash banyak digunakan untuk membangun dan memberikan efek animasi pada *website*, CD Interaktif dan yang lainnya. Selain itu aplikasi ini juga dapat

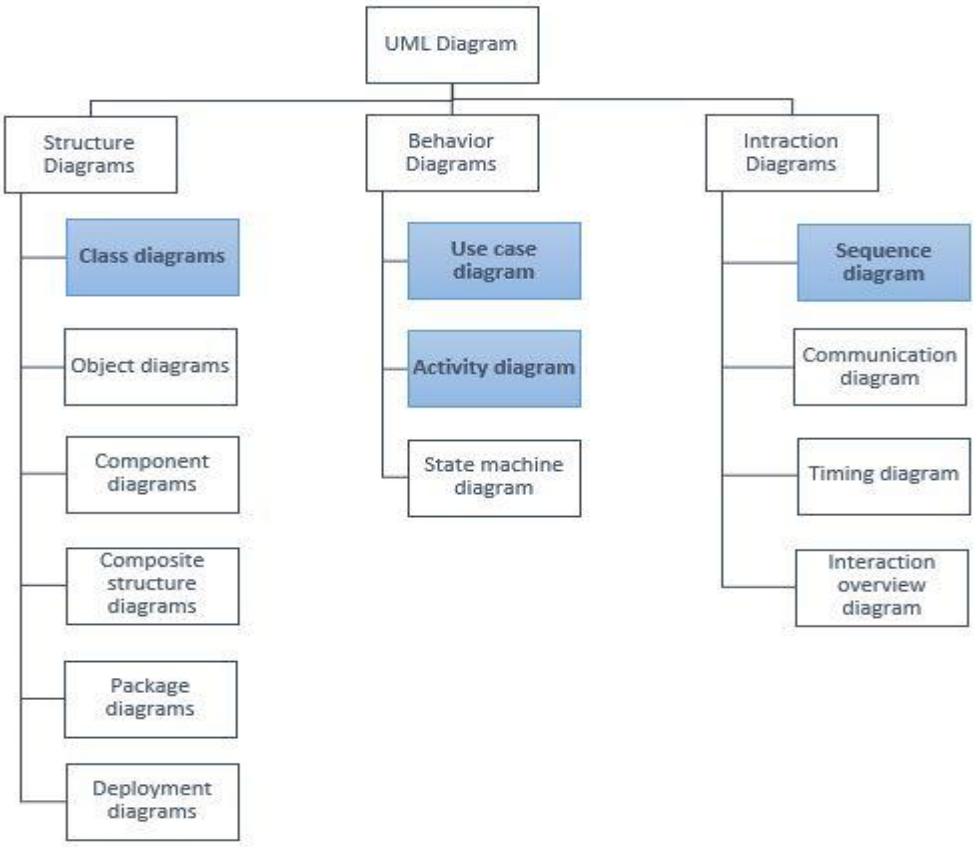
digunakan untuk membuat animasi logo, *movie*, *game*, pembuatan navigasi pada situs web, tombol animasi, *banner*, menu interaktif dan pembuatan aplikasi-aplikasi web lainnya. Dalam Flash, terdapat teknik-teknik membuat animasi, fasilitas *action script*, *filter*, *custom easing* dan dapat memasukkan video lengkap dengan fasilitas *playback* FLV. Keunggulan yang dimiliki oleh Flash ini adalah ia mampu diberikan sedikit kode pemograman baik yang berjalan sendiri untuk mengatur animasi yang ada didalamnya atau digunakan untuk berkomunikasi dengan program lain seperti HTML, PHP, dan Database dengan pendekatan XML, dapat dikolaborasikan dengan web, karena mempunyai keunggulan antara lain kecil dalam ukuran file *outputnya*.

Movie Flash memiliki ukuran file yang kecil dan dapat ditampilkan dengan ukuran layar yang dapat disesuaikan dengan keingginan. Aplikasi Flash merupakan sebuah standar aplikasi industri perancangan animasi web dengan peningkatan pengaturan dan perluasan kemampuan integrasi yang lebih baik. Banyak fitur-fitur baru dalam Flash yang dapat meningkatkan kreativitas dalam pembuatan isi media yang kaya dengan memanfaatkan kemampuan aplikasi tersebut secara maksimal. Fitur-fitur baru ini membantu kita lebih memusatkan perhatian pada desain yang dibuat secara cepat, bukannya memusatkan pada cara kerja dan penggunaan aplikasi tersebut. Flash juga dapat digunakan untuk mengembangkan secara cepat aplikasi-aplikasi web yang kaya dengan pembuatan *script* tingkat lanjut. Di dalam aplikasinya juga tersedia sebuah alat untuk men-*debug* script. Dengan menggunakan *code hint* untuk mempermudah dan mempercepat pembuatan dan pengembangan isi ActionScript secara otomatis. Untuk memahami keamanan Adobe Flash dapat dilihat dari beberapa sudut pandang, berdasarkan beberapa sumber referensi bahwa tidak ada perbedaan menyolok antara HTML dan JavaScript dimana didalamnya terdapat banyak *tools* yang dapat diambil dari SWF termasuk *ActionScript*. Sehingga kode data dapat terjamin keamanannya. Oleh sebab itu, semua kebutuhan data yang terdapat dalam SWF dapat diambil kembali melalui server. Keuntungan menggunakan metode yang sama dengan menggunakan aplikasi web yang standar adalah akan menjamin dan mengamankan penyimpanan dan perpindahan data.

2.2.21 UML (Unified Modeling Language)

Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung [11].

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek seperti pada gambar 2.15.



Gambar 2.15 UML

Melalui berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML, sebenarnya konsepsi dasar UML dapat dirangkum seperti pada tabel 2.2.

Tabel 2.2 Konsep Dasar UML

Major Area	***View***	***Diagrams***	***Main Concepts***
Structural	*Static view*	*Class diagram*	*Class, association, generalization, dependency, realization, interface*
	Use case view	*Use case diagram*	*Use case, actor, assosiation, extend, include, use case generalization*
	Implementation view	*Component diagram*	*Component, interface, dependency, realization*
	Deployment view	*Deployment diagram*	*Node, component, dependency, location*
Dynamic	*State machine view*	*Statechart diagram*	*State, event, transition, action*
	Activity view	*Activity diagram*	*State, activity, completion transition, fork, join*
	Interaction view	*Sequence diagram*	*Interaction, object, message, activation*
		Collaboration diagram	*Collaboration, interaction, collaboration role, message*
Model management	*Model management view*	*Class diagram*	*Package, subsystem, model*
Extensibility	*All*	*All*	*Constraint, stereotype, tagged values*

Berikut merupakan beberapa diagram yang ada pada UML :

1. *Use Case* Diagram
2. *Class* Diagram
3. *Activity* Diagram
4. *Sequence* Diagram

2.2.22 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang aktor sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [11].

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal.Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2.2.23 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi) [11].*Class* diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

1. Nama
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. ***Private***, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. ***Protected***, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. ***Public***, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar *Class* :

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”). Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
3. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence* diagram yang akan dijelaskan kemudian.

2.2.24 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi [11].

Activity diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity* diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu [11].

2.2.25 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait) [11].

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

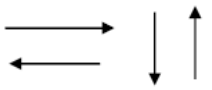
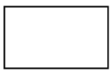
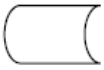
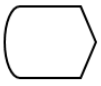
Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada *fase* desain berikutnya, *message* akan dipetakan menjadi operasi atau metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses,

biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

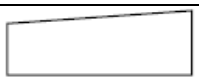
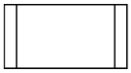
2.2.26 Flowchart

Flowchart atau bagan alir adalah bagan yang menunjukan aliran di dalam program atau prosedur sistem secara logika. *Flowchart* ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *flowchart* ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem [12]. Adapun simbol-simbol yang sering digunakan dalam *flowchart* dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol *Flowchart*

Simbol	Nama Simbol	Keterangan
	*Flow Direction symbol*	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga *connecting line*.
	*Connector Symbol*	Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.
	*Connector Symbol*	Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.
	*Processing Symbol*	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer
	*Decision Symbol*	Simbol pemilihan proses berdasarkan kondisi yang ada.
	*Disk Symbol and On-line Storage*	Simbol yang menyatakan *input* yang berasal dari *disk* atau disimpan ke *disk*.
	Simbol *Display*	Simbol yang menyatakan peralatan output yang digunakan yaitu layar, *plotter*, *printer* dan sebagainya.
	Simbol *Input-Output*	Simbol yang menyatakan proses *input* dan *output* tanpa tergantung dengan jenis peralatannya

Tabel 2.3 Simbol *Flowmap* (Lanjutan)

Simbol	Nama Simbol	Keterangan
	Simbol Manual *Input*	Simbol untuk pemasukan data secara manual *on-line keyboard*
	Simbol Manual *Operation*	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer
	Simbol *Predefine* Proses	Simbol untuk pelaksanaan suatu bagian (sub-program)/*prosedure*
	Simbol *Preparation*	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam *storage*.
	*Terminator Symbol*	Yaitu simbol untuk permulaan (*start*) atau akhir (*stop*) dari suatu kegiatan

2.2.27 Teknik Pembuatan Multitouch

Perangkat *multitouch* dapat berbeda, tergantung dari bahan dan alat yang digunakan, bahan utama yang umum digunakan adalah cahaya *infrared* dan kamera inframerah. Poin penting yang mempengaruhi hasil sentuhan adalah rata-rata penangkapan gambar per detik pada kamera dan kualitas kamera [26].

1. Sumber Cahaya Infra Merah

   Infrared (Infra Merah) adalah bagian dari spektrum cahaya yang tidak dapat dilihat oleh mata manusia. Ini adalah berbagai panjang gelombang lebih panjang dari cahaya tampak. Dalam hal *multitouch*, cahaya inframerah digunakan untuk membedakan antara gambaran visual pada permukaan sentuh, gambar 2.16 merupakan contoh LED inframerah.



Gambar 2.16 Sumber Cahaya Infra Merah

2. Kamera

Webcam Sederhana bekerja sangat baik untuk pembuatan multitouch, tetapi perlu dimodifikasi dengan mengganti lensa camera tersebut dengan lensa dengan filter inframerah, seperti gambar 2.17.



Gambar 2.17 Kamera Infra Merah

3. Permukaan Sentuhan

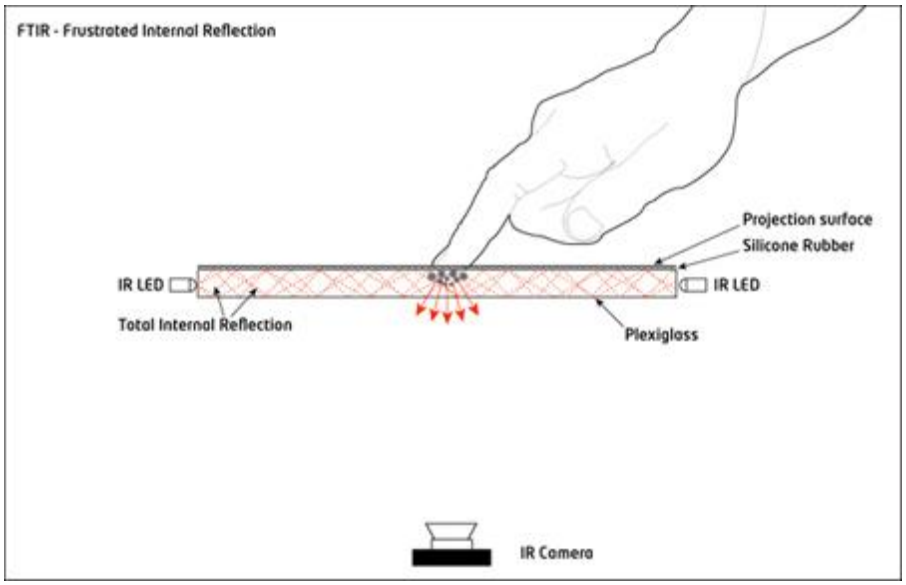
Permukaan Sentuhan bisa berbahan dasar kaca atau *acrylic* yang telah dilapis oleh layar proyeksi untuk area sentuhnya, seperti gambar 2.18.



Gambar 2.18 Permukaan Sentuhan

Ada beberapa macam teknik pembuatan *multitouch* berbasik optik diantaranya FTIR, DI, LLP, DSI dan LED-LP.

1. FTIR (*Frustrated Total Internal Reflection*)



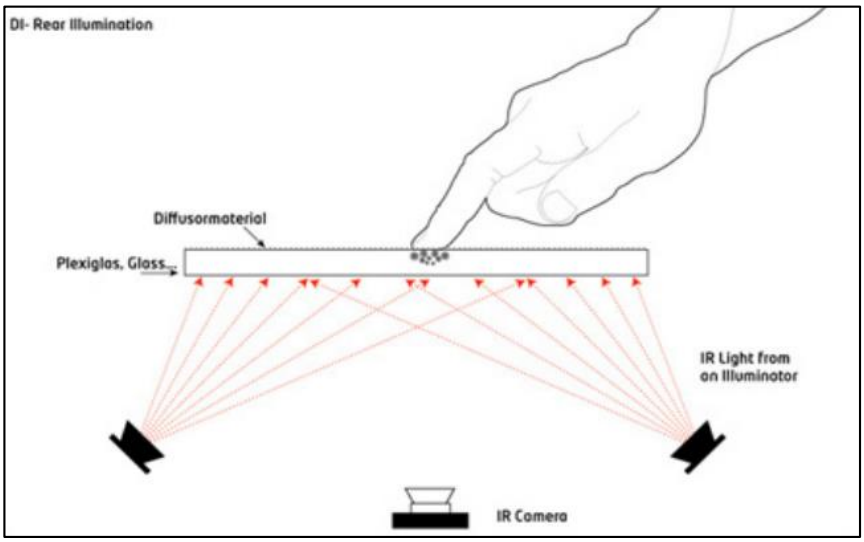
Gambar 2.19 Metode FTIR

FTIR (*Frustrated Total Internal Reflection*) adalah nama yang digunakan oleh suatu komunitas *multitouch* untuk mendiskripsikan metode yang dikembangkan oleh Jeff Han (2005). Metode yang digunakan adalah dengan meletakkan kamera-IR di bawah layar yang dapat berupa *acrylic* atau kaca yang dilapisi oleh karet silikon dan dibatasi oleh IR-LED yang berfungsi sebagai sensor seperti gambar 2.19. IR-LED memantulkan sinar sensor ke dalam lapisan layar dan berefek ketika tangan menyentuh layar. Tabel 2.4 merupakan kekurangan dan kelebihan metode FTIR.

Tabel 2.4 Kekurangan dan Kelebihan Metode FTIR

Kelebihan	Kekurangan
• Tidak memerlukan suatu Box tertutup • Deteksi memiliki kontras tinggi • Memperbolehkan jenis deteksi yang bervariasi tinggi • Dengan permukaan yang mendukung dapat digunakan pada suatu alat kecil seperti pena	• Keahlian khusus dibutuhkan untuk pemasangan beberapa tipe LED • Membutuhkan permukaan khusus (karet silikon) untuk penggunaan maksimal • Tidak dapat mengenal objek atau marker khusus

2. DI *(Diffused Illumination)*



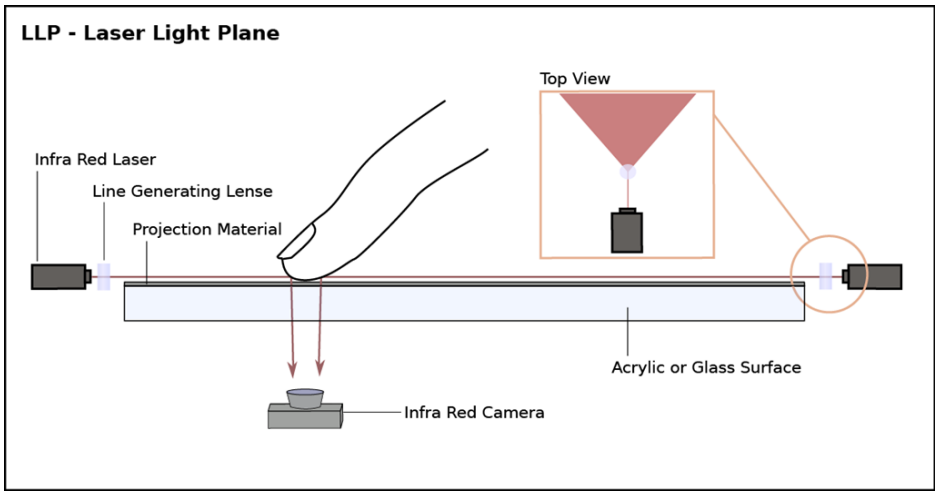
Gambar 2.20 Metode *Rear* DI

Metode lain yakni *Diffused Illumination* (DI) seperti yang terlihat pada Gambar 2.20, terbagi menjadi dua yakni *Rear*-DI dan *Front*-DI di mana keduanya berprinsip dasar yang sama yang menjadi pembeda adalah IR yang ditembakkan berposisi di bawah layar pada *Rear*-DI sedangkan untuk *Front*-DI, IR ditembakkan pada atas layar. Tabel 2.5 merupakan kekurangan dan kelebihan metode DI.

Tabel 2.5 Tabel Kekurangan dan Kelebihan Metode DI

Kelebihan	**Kekurangan**
• Tidak memerlukan permukaan khusus • Dapat menggunakan permukaan *transparant* seperti kaca (tidak harus *acylic*) • Tidak membutuhkan bingkai LED • Tidak perlu proses menyolder untuk cahaya inframerah • Dapat mendeteksi objek, tangan, titik atau gerakan	• Susah untuk mendapatkan deteksi yang bagus • Deteksi memiliki kontras rendah • Lebih berpeluang kesalahan deteksi • Dibutuhkan kotak tertutup (*Box*) untuk metode rear-DI

3. LLP *(Laser Light Plane)*



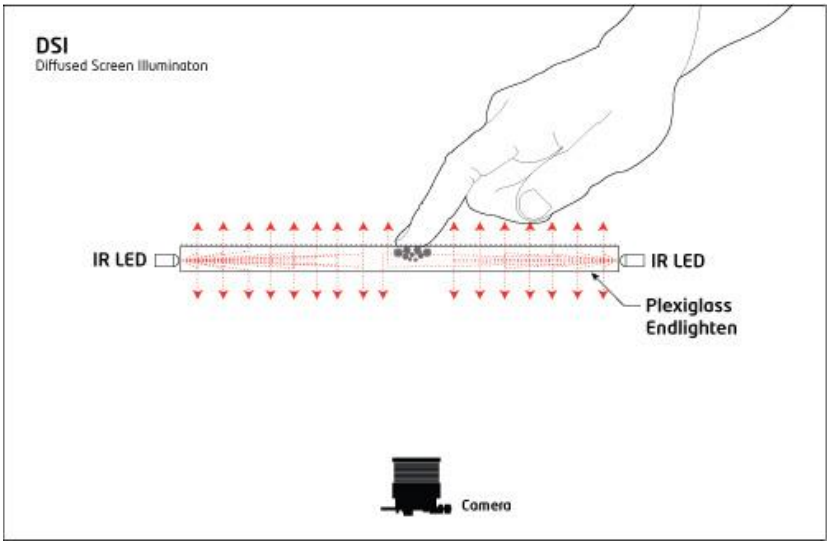
Gambar 2.21 Metode LLP

Berbeda dengan FTIR dan DI, *Laser Light Plane* (LLP), seperti yang terlihat pada Gambar 2.21 menggunakan laser-IR sebagai pengganti IR-LED. Prinsip yang digunakan lebih sederhana yakni hanya dengan menembakkan laser-IR di atas permukaan layar seperti jaring dan apabila jaring menerima sentuhan, jaring akan terputus dan mengirimkan suatu perintah ke perangkat lunak. Tabel 2.6 merupakan kekurangan dan kelebihan metode LLP.

Tabel 2.6 Tabel Kekurangan dan Kelebihan Metode LLP

Kelebihan	**Kekurangan**
• Tidak memerlukan permukaan khusus • Dapat menggunakan permukaan *transparant* seperti kaca (tidak harus *acylic*) • Tidak membutuhkan bingkai LED • Pemasangan yang mudah • Lebih murah dari teknik yang lain	• Tidak dapat mendeteksi objek dan titik. • Tidak sensitif • Lebih berpeluang kesalahan deteksi

4. DSI *(Diffused Screen Illumination)*



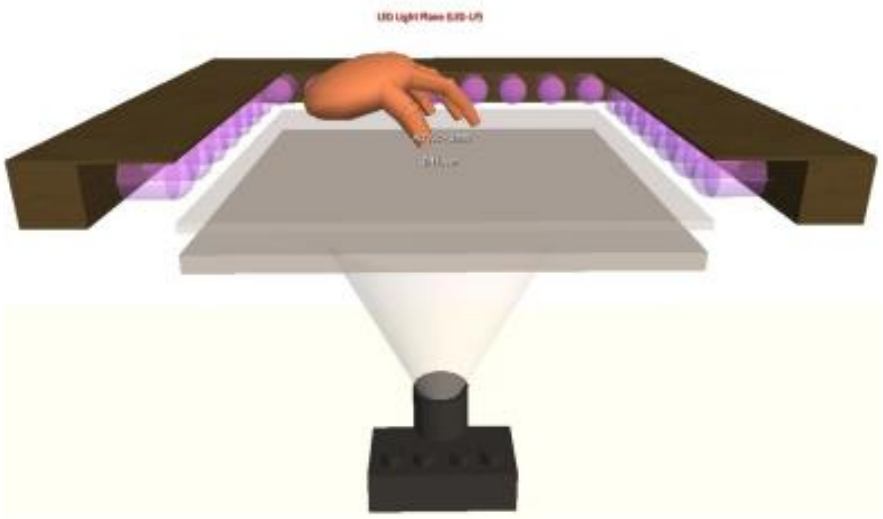
Gambar 2 22 Metode DSI

Lain halnya dengan *Diffused Surface Illumination* (DSI), seperti yang terlihat pada Gambar 2.22. DSI memakai suatu jenis acrylic yang istimewa di mana acrylic menyebarkan sensor ke atas dan ke bawah secara vertikal. Hal tersebut yang menjadi pembeda DSI dengan FTIR. Sensor yang tersebar ini bila bergesekan dengan tangan akan menjadi bentuk inputan dalam *multitouch*. Tabel 2.7 merupakan kekurangan dan kelebihan metode DSI.

Tabel 2.7 Tabel Kekurangan dan Kelebihan Metode DSI

Kelebihan	**Kekurangan**
• Memerlukan permukaan khusus • Dapat dialihkan ke metode FTIR • Dapat mendeteksi objek • Deteksi Sensitif • Memungkinkan penerangan objek	• *Acrylic* yang dibutuhkan lebih mahal bila ingin memakai permukaan silikon • Ada batasan ukuran deteksi pengaruh kelembutan permukaan • Memiliki deteksi dengan kontras rendah • Perlu keahlian khusus untuk pemasangan *infrared*

5. LED–LP *(LED Light Plane)*



Gambar 2.23 Metode *LED*-LP

Metode yang umum digunakan sekarang adalah LED *Light Plane* (LED-LP) seperti yang terlihat pada Gambar 2.23 yakni metode di mana dasar penginstalasian mirip dengan FTIR yang menjadi pembeda adalah penambahan LED pada lapisan atas layar. Tabel 2.8 merupakan kekurangan dan kelebihan metode LED-LP.

Tabel 2.8 Tabel Kekurangan dan Kelebihan Metode LED-LP

Kelebihan	**Kekurangan**
• Tidak Memerlukan permukaan khusus • Dapat menggunakan permukaan *transparant* seperti kaca (tidak harus *acrylic*) • Tidak membutuhkan bingkai LED • Tidak membutuhkan *box* • Dapat lebih murah dari teknik lain	• Pendekatan deteksi yang kurang • Tidak dapat mendeteksi objek • Hanya *Narrow-Beam LED* yang dapat dipakai

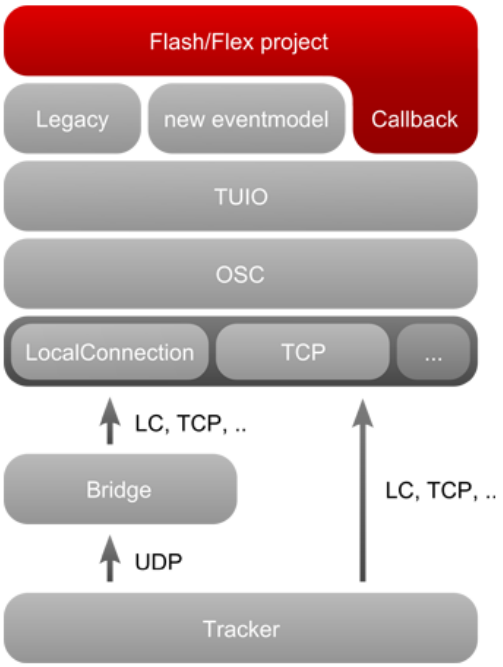
2.2.28 TUIO

TUIO *(Tangible User Interface Protocol)* adalah *framework open source* yang mendefinisikan secara umum protokol dan API untuk permukaan *multitouch*. Protokol TUIO memungkinkan transmisi yang didapat dari permukaan sentuhan termasuk *event* sentuhan dan objek yang ada pada layar sentuh [27]. Protokol ini

mengkonversi (*encode*) data dari perangkat lunak CCV/Reactivision ke aplikasi yang telah mempunyai kemampuan untuk membaca (*decoding*) data dari *protocol*.

Protokol TUIO telah diimplementasikan menggunakan *library Open Sound Control* (OSC). OSC digunakan tidak hanya untuk instrumen *musical* tetapi bisa juga digunakan untuk semua *platform* yang mendukung protokol ini.

Proses komunikasi protokol TUIO menggunakan UDP (*User* Datagram *Protocol*) agar waktu pengiriman data lebih cepat. Ketika menggunakan UDP kelemahannya adalah banyaknya data yang hilang. Oleh karena itu alternative proses komunkasi data menggunakan TCP *(Transmision Control Protocol)* akan lebih aman, tetapi waktu yang dibutuhkan lebih lama. *Flash* membutuhkan untuk mengkonversi paket UDP ke TCP. Proses konversi bisa dilakukan dengan *tool* yang bernama FLOSC (Flash OSC). Pada gambar 2.24 menggambarkan alur *communication protocols* TUIO [28].

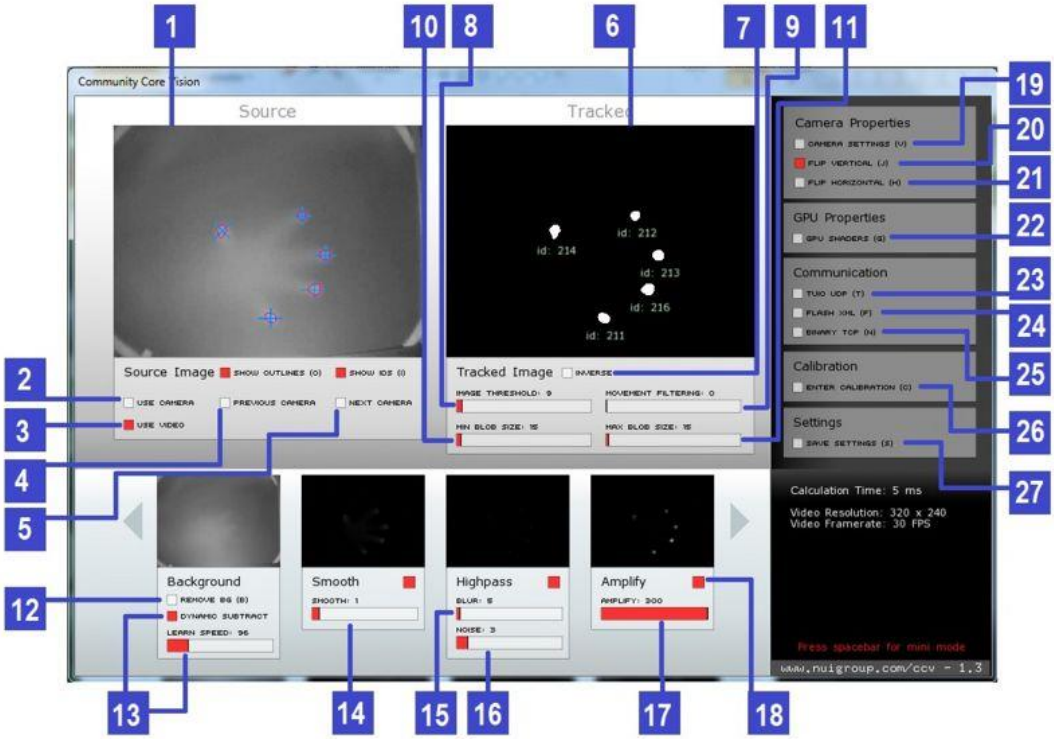


Gambar 2.24 Alur *Communication Protocols*

2.2.29 CCV (Community Core Vision)

Community Core Vision adalah perangkat lunak *open source* untuk pengolahan citra dan pengenalan pola *input* video atau *output tracking* data (contohnya koordinat dan ukuran sentuhan) dan kondisi (contohnya jari menekan,

bergerak dan mengangkat) yang dibutuhkan untuk membangun sebuah aplikasi *multitouch*. Perangkat lunak ini dapat beroperasi pada MS Windows, Linux dan Mac OS [29].



Gambar 2.25 Fitur Community Core Vision

Fitur dari perangkat lunak *Community Core Vision* dapat dilihat pada gambar 2.25. Berikut adalah keterangan dari fitur *community core vision* :

1. *Source image* - Menampilkan gambar video baik dari kamera atau file video.
2. *Use Camera Toggle* - Mengatur sumber input ke kamera dan meraih *frame* dari kamera yang dipilih.
3. *Use Video Toggle* - Mengatur sumber input untuk video dan meraih *frame* dari file video.
4. *Previous Camera Button* – Mendapatkan perangkat kamera sebelumnya yang terpasang ke komputer jika terpasang lebih dari satu kamera.
5. *Next Camera Button* - Mendapatkan perangkat kamera sebelumnya yang terpasang ke komputer jika terpasang lebih dari satu kamera.

6. *Tracked Image* - Menampilkan gambar akhir setelah penyaringan gambar yang digunakan untuk deteksi *blob*.
7. *Inverse – tracking* blob hitam bukan yang putih.
8. *Threshold Slider* - Mengatur tingkat *pixel* diterima. Pilihan yang lebih tinggi adalah, semakin besar gumpalan harus dikonversi dalam *tracking blob*.
9. *Movement filtering* - Sesuaikan tingkat jarak diterima (dalam piksel) sebelum gerakan gumpalan terdeteksi. Semakin tinggi pilihan tersebut, semakin Anda harus benar-benar menggerakkan jari Anda untuk CCV untuk mendaftarkan gerakan gumpalan.
10. *Min Blob Size* – Pengaturan untuk menyesuaikan tingkat ukuran *blob* minimum yang dapat diterima.
11. *Max Blob Size* - Pengaturan untuk menyesuaikan tingkat ukuran *blob* maksimal yang dapat diterima.
12. *Remove Background Button* – untuk menghapus gangguan yang menyebabkan *noise.*
13. *Dynamic Subtract Toggle* – untuk penyesuaian gambar yang ditangkap kamera terhadap cahaya.
14. *Smooth Slider* - memperhalus gambar dan mengurangi *noise*.
15. *Highpass Blur Slider* – untuk pengaturan ketajaman kamera dalam menangkap gambar.
16. *Highpass Noise* – mengurangi *noise* dari gambar yang ditangkap kamera.
17. *Amplify Slider* – untuk pengaturan gambar yang kurang jelas agar kamera dapat menangkap gambar lebih jelas.
18. *On/Off Toggle* - Digunakan pada setiap filter, ini digunakan untuk menghidupkan setiap proses filtering atau mematikan.
19. *Camera Settings Button* - Membuka pengaturan kamera.
20. *Flip Vertical Toggle* – menyesuaikan gambar dari kamera secara vertical.
21. *Flip Horizontal Toggle* – menyesuaikan gambar dari kamera secara horizontal.
22. *GPU Mode Toggle* – memproses melalui *graphic art.*

23. *Send UDP Toggle* - mengaktifkan pengiriman data melalui TUIO.
24. *Flash XML* – mengaktifkan pengiriman data ke aplikasi *flash*.
25. *Binary TCP* – mengaktifkan pengiriman pesan RAW (x,y, koordinat).
26. *Enter Calibration* – memuat layar kalibrasi.
27. *Save Settings* –menyimpan semua pengaturan saat ini ke dalam *file* xml.
28. *Image Thresholding* – untuk membedakan mana yang termasuk obyek dan *background* dari citra secara jelas.

2.2.30 Reactivision

Reactivision adalah perangkat lunak *open source* untuk pengolahan citra dan pengenalan pola *input* video atau *output tracking* data (contohnya koordinat dan ukuran sentuhan) dan kondisi (contohnya jari menekan, bergerak dan mengangkat) yang dibutuhkan untuk membangun sebuah aplikasi *multitouch*. Perangkat lunak ini dapat beroperasi pada MS Windows, Linux dan Mac OS [30].



Gambar 2.26 Fitur Reactivision

Fitur dari perangkat lunak *reactivision* dapat dilihat pada gambar 2.26. Berikut adalah keterangan dari fitur *reactivision* :

1. *N* - Berfungsi untuk menonakifkan hasil dari *capture* kamera.
2. *S* - Berfungsi untuk menampilkan citra asli yang ditangkap kamera.
3. *T* - Berfungsi untuk menampilkan hasil dari pengolahan citra dari *reactivison*.
4. *H* - Berfungsi untuk menampilkan bantuan untuk mengakses fitur yang ada di *reactivision*
5. *F1* - Berfungsi untuk menampilkan *reactivision* pada *mode fullscreen.*
6. *ESQ* - Berfungsi untuk keluar dari aplikasi *reactivision.*
7. *V* - Berfungsi untuk menampilkan proses yang dilakukan *reactivision.*
8. *O* - Berfungsi untuk menampilkan *form* pengaturan kamera.
9. *P* - Berfungsi untuk menghentikan sementara proses yang sedang dilakukan.
10. *E* - Berfungsi untuk mengambil citra yang akan digunakan untuk citra referensi.
11. *Space* - Berfungsi untuk mengaktifkan citra referensi yang telah diambil (mengaktifkan proses *background subtraction*).
12. *G* - Berfungsi untuk mengatur kedetailan pemisahan objek dengan latar belakang.
13. *F* – Berfungsi untuk menampilkan konfigurasi sentuhan
14. *C* - Berfungsi untuk menampilkan *grid calibration.*
15. *Q* - Berfungsi untuk melakukan kalibrasi secara cepat.
16. *J* - Berfungsi untuk memindahkan titik *grid* yang akan dirubah.
17. *K* - Berfungsi untuk mengembalikan posisi *point grid* ke posisi *default*.
18. *L* - Berfungsi untuk membalikan hasil kalibrasi *grid* yang telah dilakukan (*Revert Mode*).
19. *R* - Berfungsi untuk menampilkan hasil kalibrasi *grid*.
20. *A,D,W,X* – berfungsi untuk memindahkan, menggeser titik *grid* yang dipilih.

2.2.31 Penelitian Teknologi Multitouch Berbasis Optik

Banyak penelitian sebelumnya yang telah dilakukan mengenai teknologi *multitouch*. Dalam upaya pengembangan teknologi ini perlu dilakukan studi pustaka sebagai salah satu dari penerapan metode penelitian yang akan dilakukan. Diantaranya adalah mengidentifikasi metode dan teknik yang pernah dilakukan sebelumnya. Beberapa Literatur Review tersebut adalah sebagai berikut :

1. Penelitian dilakukan oleh Rizki Fauzian dari Universitas Komputer Indonesia, Bandung Indonesia tahun 2014, dengan judul "*Multitouch Digital Poster*". Penelitian ini membahas mengenai implementasi digital poster dengan konten berupa teks, gambar dan video, sehingga dalam penyampaiannya lebih interaktif dan meminimalisir penggunaan tempat. Teknik yang digunakan dalam pembuatan *Multitouch Digital poster* ini, menggunakan teknik/ metode *Laser Light Plane* (LLP). Pada metode *Laser Light Plane* (LLP), berkas sinar inframerah digunakan di atas layar dan diarahkan sejajar dengan permukaan layar *multitouch*. Sentuhan jari pada layar akan memantulkan berkas sinar inframerah tersebut ke belakang layar dan akhirnya akan ditangkap oleh kamera. Pada gambar 2.27 merupakan aplikasi dari *multitouch digital poster* [31].



Gambar 2 27 Aplikasi *Multitouch Digital Poster*

2. Penelitian dilakukan oleh Chandra Tresnadi dari Institut Teknologi Bandung, Bandung Indonesia tahun 2008, dengan judul *Game* Batik "Nitiki" Berteknologi *Multitouch screen*. Penelitian ini membahas mengenai penggabungan dunia batik (budaya tradisi) terhadap perkembangan teknologi komputer melalui media game digital. Konsep dari *game* dirancang dengan metoda *participatory* (berhubungan dengan partisipasi, keterlibatan), yaitu melibatkan emosi, indera (mata, telinga, raba, naluri), dan aktifitas fisik pemain yang menjadi faktor penting dalam membangun interaktivitas *game*play terhadap pemain dan lingkungan, pada gambar 2.28 merupakan simulasi dari *game nitiki* [32].



Gambar 2.28 Simulasi *Game Nitiki*

Teknik yang digunakan dalam pembuatan *Game* ini, menggunakan teknik *Frustrated Total Internal Reflection* (FTIR). Pada metode *Frustrated Total Internal Reflection* (FTIR), berkas sinar inframerah digunakan di tengah layar. Sentuhan jari pada layar akan memantulkan berkas sinar inframerah tersebut ke belakang layar dan akhirnya akan ditangkap oleh kamera.

3. Penelitian dilakukan oleh Alissa N. Antle, Nima Motamedi, Karen Tanenbaum, Zhen (Lesley) Xie dari Simon Faser University, Canada tahun 2009, dengan judul *The EventTable Technique: Distributed Fiducial Marker*. Penelitian ini membahas mengenai implementasi *Fiducial marker* pada teknologi *multitouch*. *Game Jigsaw Puzzle* merupakan implementasi dari penelitian ini, dimana beberapa *marker* dibuat menjadi *puzzle* yang saling berhubungan, pada gambar 2.29 merupakan bentuk *marker* dari *game jigsaw puzzle* [33].



Gambar 2.29 Bentuk *marker*