

BAB II

LANDASAN TEORI

2.1 Webcam

Kamera web (singkatan dari *web* dan *camera*) merupakan sebuah media yang berorientasi pada *image* dan video dengan resolusi tertentu. Umumnya *webcam* adalah sebuah perangkat komputer sebagai media input satu arah, yaitu berfungsi untuk masukan *image* atau video secara langsung. Dimana *camera web* (*webcam*) biasanya selalu dihubungkan melalui USB pada komputer atau laptop.



Gambar 2.1. Webcam HD (resolusi 1280x720)

2.2 Single face vs Multiple face

Single face merupakan citra wajah yang mana proses pengenalan wajah dilakukan secara otomatis terhadap objek wajah dengan mendeteksi objek hanya satu wajah (Yang, 2002). Penerapan *single face* saat ini banyak digunakan dalam absensi pengenalan wajah. Adapun keuntungan dari penerapan *single face* dalam

sistem absensi pengenalan wajah yaitu sistem hanya fokus terhadap satu objek yang terdeteksi dan hanya mengolah satu citra image yang merupakan titik acuan pengenalan. Sebaliknya kekurangan dari sistem *single face* adalah kurangnya efisiensi waktu terhadap banyak objek, sistem hanya bisa mendeteksi satu objek wajah dalam jangkauan kamera meskipun ada banyak objek .

Sedangkan *multiple face* merupakan citra wajah dimana sistem dapat mengenali banyak wajah secara otomatis. Dengan diterapkannya sistem ini banyak keuntungan yang dapat membuat sistem absensi pengenalan wajah menjadi lebih efisien, karena dapat mendeteksi dan mengenali objek wajah dalam satu kali input wajah (Rojas, 2015). Penerapan *multiple face* sangat berguna untuk absensi yang memiliki member yang cukup banyak, maka dari itu sistem *multiple face* ini dapat mendeteksi banyak objek muka yang berada dalam jangkauan kamera.

2.3 Face Detection menggunakan Haar Cascade Classifier

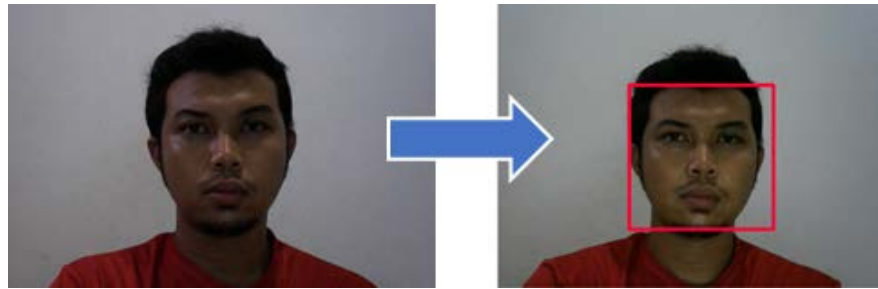
Face Detection adalah teknologi komputer yang digunakan dalam berbagai aplikasi yang mengidentifikasi wajah manusia dalam gambar digital. Deteksi wajah juga merupakan salah satu tahap praproses yang sangat penting di dalam sistem pengenalan wajah yang digunakan untuk sistem biometrik.

Deteksi wajah juga dapat digunakan untuk pencarian dan pengindeksan citra atau *video* yang di dalamnya terdapat wajah manusia dalam berbagai ukuran, posisi, dan latar belakang. Bagaimana sistem deteksi wajah ini memproses gambar

dari obyek bergerak. Pemrosesan gambar ini bertujuan untuk mencari wajah dari gambar obyek bergerak yang telah di-*capture*, kemudian gambar tersebut diolah dengan memisahkan gambar dengan latar belakangnya, sehingga hanya bagian yang dianggap kulit yang ditampilkan sedangkan bagian yang bukan kulit akan dihitamkan. Pemisahan gambar dengan latar belakang ini bertujuan untuk memudahkan proses pencarian wajah.

Deteksi objek wajah menggunakan *Haar Cascade classifier* yang merupakan sebuah fungsi pendeteksian objek wajah yang dimiliki oleh aplikasi OpenCV. *Face detector* akan melakukan proses pengujian tiap lokasi *image* dan mengklasifikasi sebagai wajah dan bukan wajah. Klasifikasi wajah ini memakai suatu ketetapan skala, misalnya 100×100 *pixel*. Jika wajah pada objek *image* lebih besar atau lebih kecil dari *pixel* tersebut, *classifier* terus menerus jalan beberapa kali, untuk mencari wajah pada gambar tersebut.

Classifier memakai data yang disimpan pada file XML untuk memutuskan bagaimana mengklasifikasi setiap lokasi *image*. OpenCV memakai 4 data XML untuk deteksi wajah depan dan wajah *profile*. Termasuk juga 3 file XML untuk bukan wajah, satu untuk mendeteksi badan secara penuh, satu untuk badan bagian atas, dan satu untuk badan bagian bawah. Kita harus memberitahukan (mendeklarasikan) letak dari *classifier* yang dipakai. Salah satunya bernama "*haarcascade_frontalface_default.xml*". Dari fungsi tersebut menghasilkan sebuah nilai *eigen* yang menjadi acuan pengenalan wajah dan menghasilkan pendeteksian objek wajah seperti pada gambar berikut.



Gambar 2.2. *Face Detection*

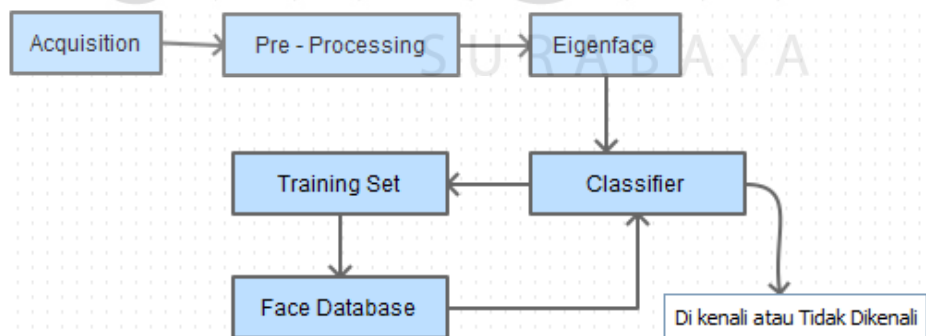
2.4 Multi Face Recognition

Face recognition adalah salah satu teknik *biometric* yang memungkinkan komputer atau mesin autentik untuk mengenal wajah manusia. Salah satu aplikasi dari pengenalan wajah adalah pengenalan banyak wajah, yaitu pengenalan wajah dari suatu citra yang terdiri dari banyak wajah. Untuk membedakan beberapa wajah manusia dalam suatu citra sangat sulit bagi sistem pengenalan wajah. Selain itu juga terdapat beberapa masalah Pada sistem pengenalan wajah otomatis, yaitu ekspresi wajah, iluminasi atau pencahayaan dan jarak. Sistem pengenalan banyak wajah merupakan salah satu solusi dari permasalahan ini.

Face Recognition atau pengenalan wajah adalah suatu sistem atau metode yang berorientasi pada wajah. Pengenalan ini dapat dibagi menjadi dua bagian yaitu : “dikenali” dan “tidak dikenali”, setelah dilakukan dengan pola yang sebelumnya disimpan di dalam database. Sistem ini juga harus bisa mengenali objek bukan wajah .

Secara umum sistem pengenalan citra wajah dibagi menjadi dua jenis, yaitu sistem *feature-based* dan sistem *image-based*. Pada sistem pertama digunakan ciri yang dieksekusi dari komponen citra wajah seperti mata, hidung, mulut, dan lain-lain yang kemudian dimodelkan secara geometris hubungan antara ciri-ciri tersebut. Sedangkan pada sistem kedua menggunakan informasi mentah dari pixel citra yang kemudian direpresentasikan dalam metode tertentu, misalnya pada *Principal Component Analysis* (PCA) atau Transformasi *Wavelet* yang digunakan untuk klasifikasi identitas citra.

Multi face recognition mengambil karakteristik alami yang ada pada tiap-tiap wajah untuk dikenali. Ada tiga tahap untuk melakukan *face recognition*, yaitu deteksi wajah, ekstraksi ciri dan klasifikasi. Deteksi wajah adalah suatu langkah dalam *face recognition* untuk menemukan posisi wajah dari sebuah citra yang akan di ekstraksi selanjutnya. Ekstraksi ciri adalah langkah untuk menentukan karakteristik alami dari suatu wajah yang selanjutnya akan diklasifikasi atau dikenali.



Gambar 2.3. Diagram Blok Pengenalan wajah

Digram blok diatas merupakan tahap sistem dalam melakukan pengenalan wajah. Perolehan (*acquisition*) untuk mendapatkan input gambar (*face image*) selanjutnya *Pre_processing* pengolahan citra *image* sebagai tahap awal pemrosesan objek wajah menghasilkan *normalized face image*, dan selanjutnya akan di lakukan proses *feature extractor* untuk ekstraksi ciri pada setiap *image* yang telah masuk untuk mendapatkan ciri yang membedakan setiap objek wajah. *Training sets* melakukan perhitungan terhadap objek wajah terhadap database yang sudah ada (*face database*). Selanjutnya dilakukan *classifier* dimana proses ini melakukan tahap yang paling penting yaitu menjalankan apakah objek wajah tersebut dianggap dikenali atau tidak.

2.5 Metode Eigenface

Eigenface berasal dari prefiks bahasa Jerman “eigen”, yang berarti “sendiri/individual”. Metode *Eigenface* dianggap sebagai teknologi pengenalan wajah otomatis pertama yang pernah diciptakan. Teori ini dikembangkan oleh Turk dan Petland.

Teori ini dikembangkan dengan membagi sebuah citra wajah menjadi data set fitur karakteristik yang disebut *eigenface*. *Eigenface* adalah salah satu algoritma pengenalan pola wajah yg berdasarkan pada *Principle Component Analysis* (PCA) yg dikembangkan di MIT. *Eigenface* merupakan kumpulan dari *eigenvector* yang dipakai untuk masalah *computer vision* pada pengenalan wajah manusia.

Fitur karakteristik merupakan komponen utama (*principal component*) dari *training* set awal dari citra wajah. Penelitian yang dilakukan oleh Carey dan Diamond menunjukkan bahwa fitur wajah sebuah individu dan hubungan langsung antar fitur tersebut tidak dapat menyamai kemampuan manusia dalam memperhatikan dan mengenal wajah. *Eigenface* adalah sekumpulan *standardize face ingredient* yang diambil dari analisis statistik dari banyak gambar wajah (Layman dalam Al Fatta, Hanif, 2009).

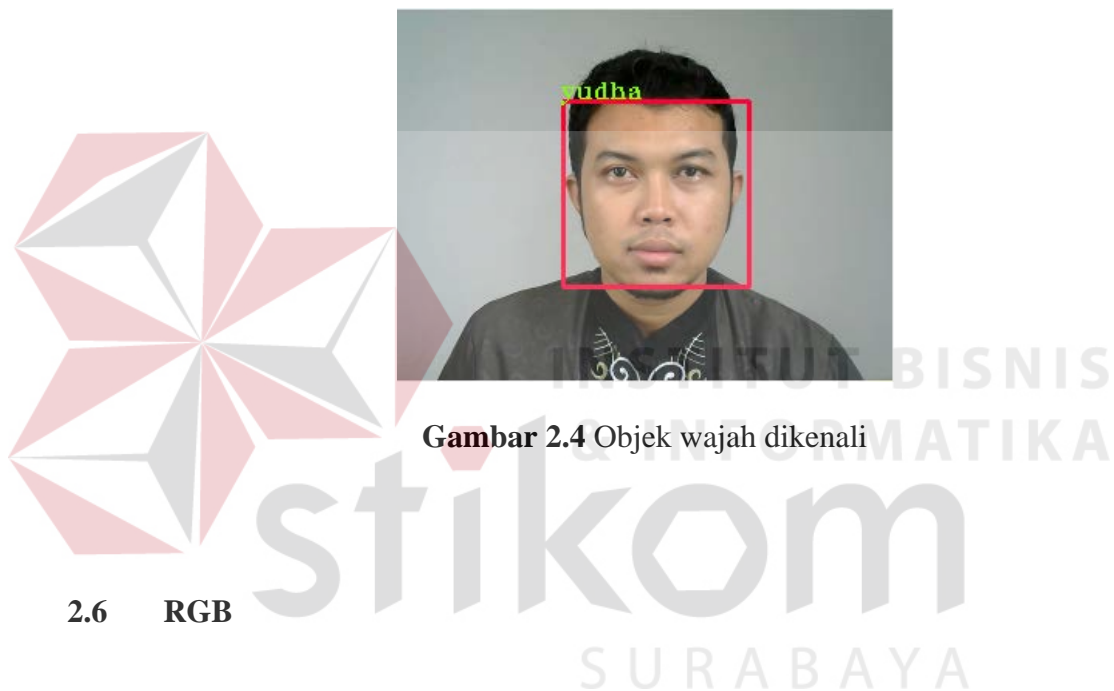
Untuk menghasilkan *eigenface*, sekumpulan citra digital dari wajah manusia diambil pada kondisi pencahayaan yang sama kemudian dinormalisasikan dan diproses pada resolusi yang sama (misal $m \times n$), kemudian citra tadi diperlakukan sebagai vektor dimensi $m \times n$ dimana komponennya diambil dari nilai piksel citra.

Metode *eigenface* menggunakan *eigenvectors* dan *eigenvalues* untuk mempresentasikan *image* wajah. *Eigenvector* merupakan sekumpulan dari fitur yang mencirikan variasi antara *image* wajah. Setiap lokasi *image* wajah memberikan kontribusi yang lebih atau kurang untuk setiap *eigenvector*, sehingga tampilan dari *eigenvector* seperti wajah yang samar. Dalam mengenali objek wajah, hasil deteksi wajah kemudian dibandingkan dengan wajah yang tersimpan di *database/template*.

2.5.1 Konsep Eigenface

1. Menghitung jarak dari gambar tersebut dibandingkan dengan gambar-gambar yang ada di dalam *database*. (menggunakan perhitungan *Euclidean Distance*)

2. Memilih sebuah gambar dari *database* yang mendekati wajah yang ada di dalam gambar tersebut.
3. Jika jarak yang telah diukur tersebut hasilnya mendekati nilai dari *threshold* maka gambar tersebut dikenali oleh sistem, tetapi bila nilai yang dihasilkan memiliki perbedaan yang jauh maka gambar tersebut termasuk dalam gambar yang tidak dikenali oleh sistem karena sistem hanya mengenali gambar yang ada di dalam database.



Gambar 2.4 Objek wajah dikenali

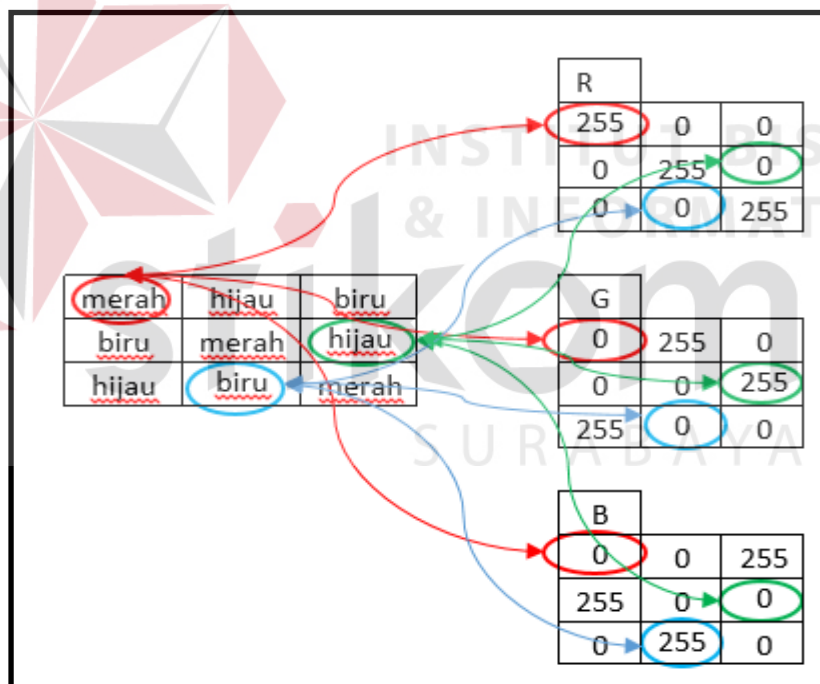
2.6 RGB

RGB adalah suatu komponen penyusun gambar yang mana terdiri atas 3 warna yaitu merah(*Red*) R, hijau(*Green*) G, dan biru(*Blue*) B, yang menjadi patokan secara *universal*. Dari ketiga warna tersebut dikombinasikan menjadikan warna – warna pada setiap pixel sehingga menjadi gambar. Setiap piksel pada warna membutuhkan citra data sebesar 3 byte. Setiap warna dasar memiliki intensitas unuk menentukan seberapa besar tingkatan warna tersebut dengan nilai

nol (0) dan nilai maksimum 255 (8 bit). Dalam RGB digunakan kode – kode untuk menentukan warna. Seperti gambar berikut :



Gambar 2.5 Representasi Warna Pada RGB



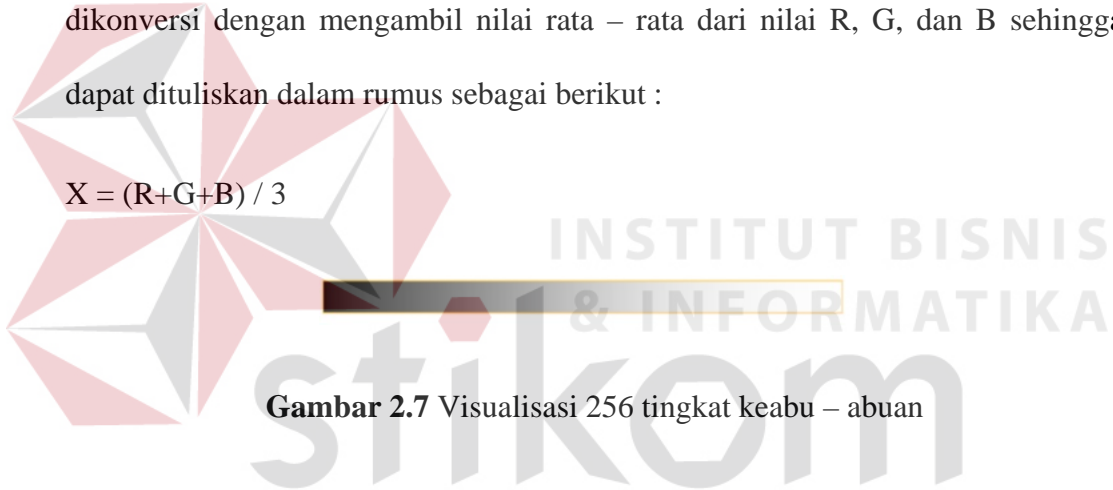
Gambar 2.6 Komponen Gambar (RGB)

2.7 Mengubah Gambar Berwarna Menjadi *Grayscale*

Grayscale adalah warna – warna dalam *pixel* yang berada dalam rentang gradasi warna hitam dan putih. Citra yang ditampilkan dari *grayscale* adalah citra yang terdiri atas warna abu – abu, berorientasi pada warna hitam pada gradasi warna dengan intensitas lemah dan warna putih pada gradasi dengan intensitas warna kuat.

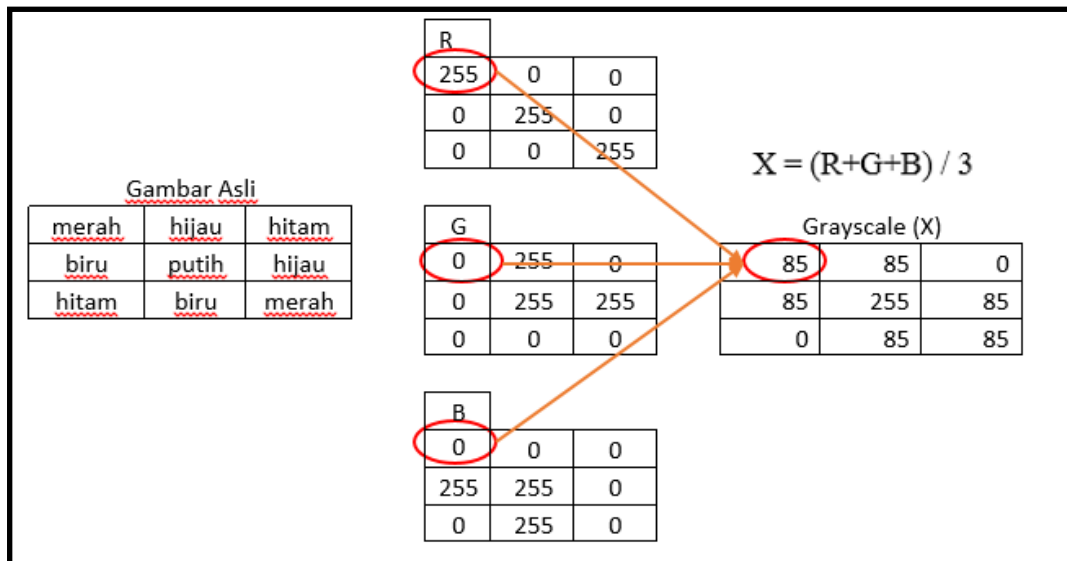
Citra *grayscale* memiliki format 8 bit pada setiap piksel, yang memiliki intensitas tingkatann warna sebanyak 256 tingkatan. Untuk mengubah citra berwarna dalam suatu *image* R,G, dan B menjadi *image* grayscale, maka dapat dikonversi dengan mengambil nilai rata – rata dari nilai R, G, dan B sehingga dapat dituliskan dalam rumus sebagai berikut :

$$X = (R+G+B) / 3$$



Gambar 2.7 Visualisasi 256 tingkat keabu – abuan

Dengan rumus tersebut maka dapat di implementasikan bahwa seperti contoh pada gambar 2.12. proses gambar berwarna diubah menjadi *grayscale* sebagai berikut :



Gambar 2.8 konversi *Image* Berwarna menjadi Grayscale

Dari proses tersebut selanjutnya akan di proses menggunakan fungsi *library* “*haarcascade_Classifier.xml*” dalam hal ini klasifikasi pada *frontal face* yaitu “*haarcascade_frontalface_default.xml*,” sebagai pendeteksi objek wajah tampak depan lurus. Pada penggunaan *library* ini menghasilkan sebuah nilai yang menjadi acuan terhadap pengenalan wajah.

2.8 *Euclidean Distance*

Euclidean Distance merupakan perhitungan yang paling sering digunakan untuk menghitung jarak antara 2 vektor. Cara perhitungan *Euclidean Distance* adalah dengan menghitung akar dari kuadrat perbedaan 2 vektor (jarak antara 2 titik).

Rumus *Euclidean Distance* sebagai berikut :

$$d_{ab} = \sqrt{\sum_{k=1}^n (X_{ak} - X_{bk})^2}$$

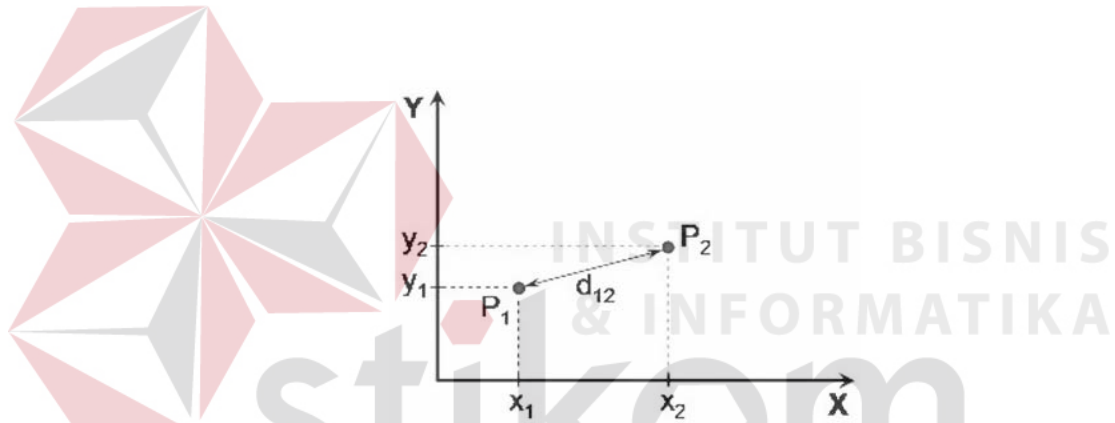
Keterangan :

d_{ab} : *Euclidean Distance*

X : nilai vektor

ak, bk : vektor yang dioperasikan

contoh : *image* diketahui mempunyai vektor $P_1 = (213)$, $Y = (124,4)$.



Gambar 2.9 Jarak Euclidean Sumber : (<http://www.metode-algoritma.com/2013/06/pca-untuk-pengenalan-wajah.html>)

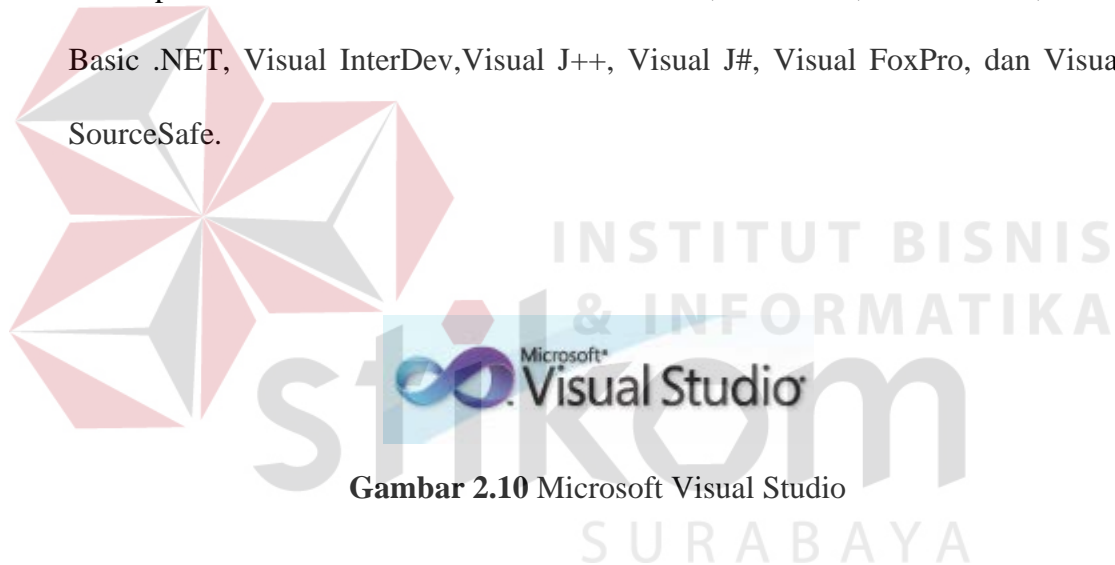
Penyelesaian : $P_1 = (213)$, $Y = (124,4)$.

$$\begin{aligned} \text{Euclidean Distance } d_{12} &= \sqrt{(213 - 124)^2 + (0 - 4)^2} \\ &= \sqrt{(89)^2 + 0} \\ &= \sqrt{7921} \\ &= 89 \end{aligned}$$

Sehingga jarak *euclidean* dari kedua titik P_1 dan P_2 adalah 89.

2.9 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.



Gambar 2.10 Microsoft Visual Studio

2.10 SQL Server

SQL Server merupakan sebuah sistem untuk manajemen database relasional yang dirancang untuk aplikasi dengan arsitektur *client/server*. SQL adalah singkatan dari *Structured Query Language* dimana *client*, *server*, dan *client/server* dapat digunakan untuk merujuk kepada konsep yang sangat umum dan lebih spesifik terhadap perangkat keras atau perangkat lunak (DionMustofa, 2013)



2.10.1 Pengertian Client dan Server

Client adalah setiap komponen penerima (meminta layanan atau sumberdaya dari suatu sistem). Sedangkan *Server* adalah setiap komponen yang menyediakan layanan atau sumberdaya ke komponen sistem lainnya. Sistem *client server* ini merupakan sebuah sistem dimana salah satu mengirim *request* dan yang satunya menunggu *request*. Dalam hal ini sebuah server selalu menunggu *request* dari *client* untuk mendapatkan layanannya. Sedangkan *client* akan mengeluarkan *request* untuk mendapatkan layanan dari *server*.

2.10.2 Kegunaan SQL

Berikut merupakan kegunaan SQL :

1. Memungkinkan untuk mengakses dan memanipulasi database.

2. Dapat mengeksekusi *query* terhadap database..
3. Dapat mengambil data dari database.
4. Dapat menyisipkan catatan dalam database.
5. Dapat membuat database baru.
6. Dapat membuat tabel baru dalam database.
7. Dapat membuat prosedur yang tersimpan dalam database.
8. Dapat mengatur hak akses pada tabel, prosedur, dan pandangan.

