10:43 PM



# PEMBANGUNAN APLIKASI SIMULATOR KECAPI SEBAGAI MEDIA PROMOSI KEBUDAYAAN BERBASIS ANDROID STUDI KASUS DINAS PARIWISATA DAN KEBUDAYAAN PROVINSI JAWA BARAT


Oleh :

RISNANDAR

10108026








Penguji 1: Andri Heryandi, S.T., M.T.
Penguji 2: Ir. Taryana Suryana, M.Kom.
Penguji 3: Irfan Maliki, S.T.,M.T.

# Latar Belakang Masalah

- Dinas Pariwisata dan Kebudayaan Provinsi Jawa Barat merupakan salah satu Lembaga Daerah di lingkungan Pemerintah Provinsi Jawa Barat yang memiliki tugas pokok sebagai penyelenggaraan pelestarian, pemanfaatan dan pengembangan kebudayaan.
- Seni budaya Indonesia terutama alat musik tradisional saat ini mulai terlupakan.
- Kurang populernya alat musik kecapi di kalangan generasi muda.

# Latar Belakang Masalah (2)

- Semakin sedikitnya orang yang mampu memainkan alat musik kecapi, membuat kurangnya penyebaran informasi kecapi yang merupakan alat musik petik kebudayaan Jawa Barat.
- Diperlukan kepekaan tinggi terhadap nada kecapi untuk menyelaraskan nada kecapi.
- Diperlukannya suatu hiburan yang berkualitas, sarana-sarana yang meningkatkan kreatifitas yang mengangkat khasanah budaya.

# Identifikasi Masalah

"Bagaimana cara membangun aplikasi simulator kecapi sebagai media promosi kebudayaan Provinsi Jawa Barat berbasis android"

# Maksud dan Tujuan

- **Maksud**
  - Maksud dari pembuatan tugas akhir ini adalah membangun aplikasi simulator kecapi pada *platform* android
- **Tujuan**
  - Memperluas informasi alat musik kecapi sebagai kebudayaan Jawa Barat.
  - Mempopulerkan alat musik kecapi di kalangan generasi muda.
  - Membantu dalam proses penyeteman nada pada kecapi.
  - Memberikan sarana untuk menuangkan ide dalam permainan kecapi melalui sarana perekaman permainan kecapi.

# Batasan Masalah

- Aplikasi simulator kecapi yang dibangun berbasis mobile dengan sistem operasi Android versi 2.1.
- Titi laras yang digunakan yaitu titi laras pelog degung dan madenda.
- Perekaman nada bersifat statis, hanya menggunakan 1 tempat penyimpanan.
- Lagu tutorial yang disediakan pada aplikasi hanya 2 lagu.
- Model analisis sistem adalah pemodelan berorientasi objek dan peralatan yang digunakan adalah UML (*Unified Modeling Language*).

10:43 PM

# kecapi

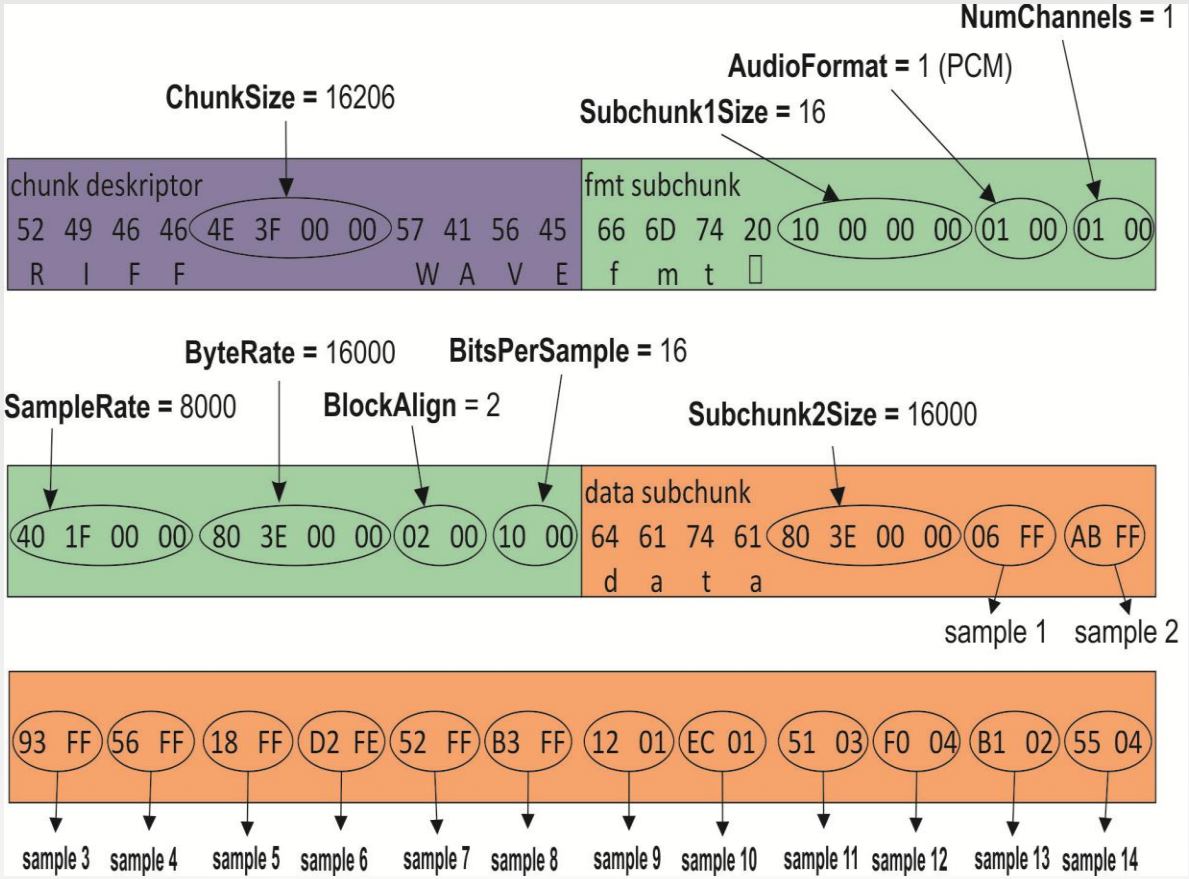


- Kecapi adalah alat musik petik yang berasal dari daerah Jawa Barat. Bentuk organologi kecapi adalah sebuah kotak kayu yang diatasnya berjajar dawai/senar, kotak kayu tersebut berguna sebagai resonatornya.
- Sekitar tahun 1950-an ada seniman terkenal, *Mang Koko*, yang membuat kacapi dengan dawai yang berjumlah 20.
- Kacapi ini kadang disebut kacapi modern, kadang disebut *kacapi wanda anyar* ("wajah baru").
- Kacapi ini kemudian sangat populer, terutama karena diajarkan di sekolah kesenian yang pertama di Jawa Barat, Konservatori Karawitan, di mana Mang Koko adalah juga direkturnya (yang kedua, setelah *Machyar Kusumadinata*).

# Analisis Frekuensi

- Frekuensi nada kecapi ini diperoleh dari hasil analisis dari data hasil perekaman nada kecapi yang dicuplik dengan frekuensi 8000 Hz selama 1 detik yang disimpan dengan format wav

# Analisis Frekuensi (2)

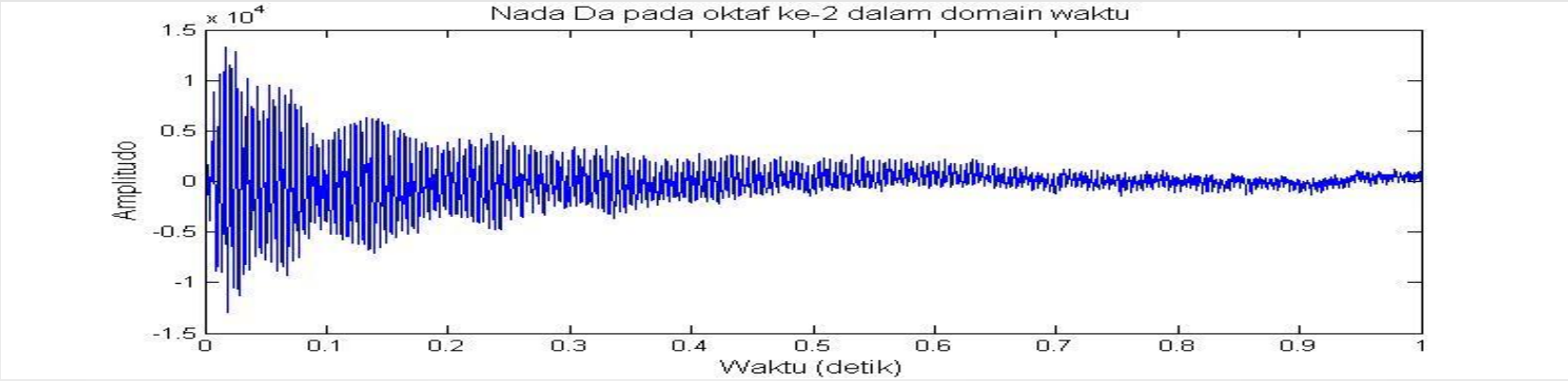



Tabel. Data Wav (desimal)

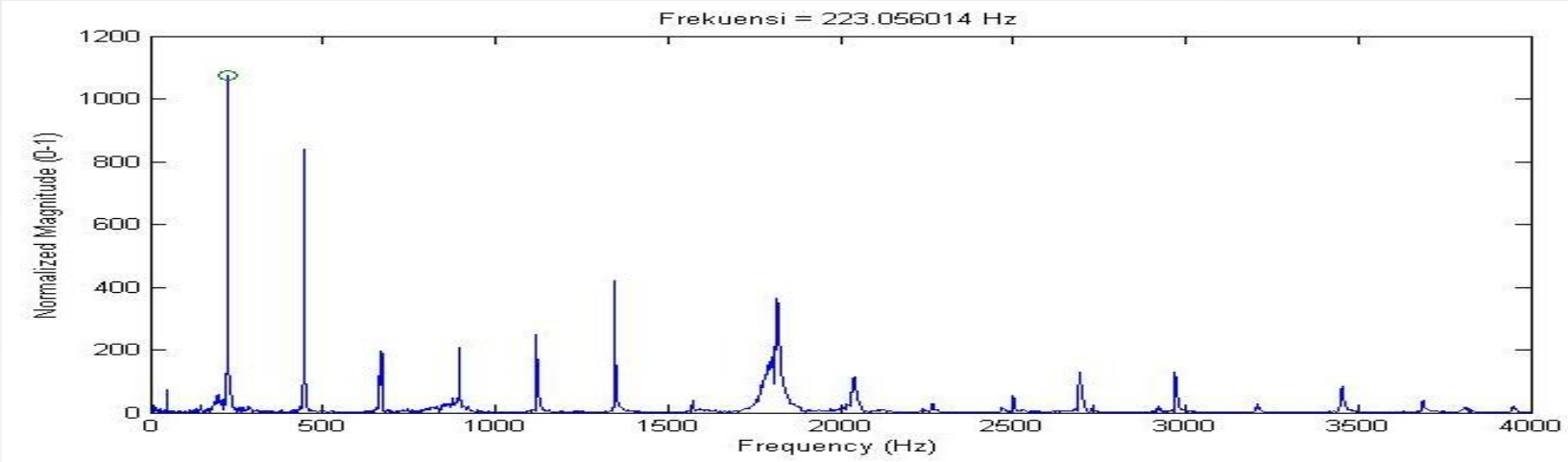
| Sample | Channel |
|---|---|
| 1 | -250 |
| 2 | -85 |
| 3 | -109 |
| 4 | -170 |
| 5 | -232 |
| 6 | -302 |
| 7 | -174 |
| 8 | -77 |
| … | … |
| 8000 | -607 |

Gambar . Intepretasi File Audio *wav* nada “Da” kecapi pada oktaf kedua

# Analisis Frekuensi (3)




Gambar File Audio *wav* nada "Da" kecapi pada oktaf kedua dalam domain waktu.




Gambar File Audio *wav* nada "Da" kecapi pada oktaf kedua dalam domain frekuensi.

# Analisis Frekuensi (4)

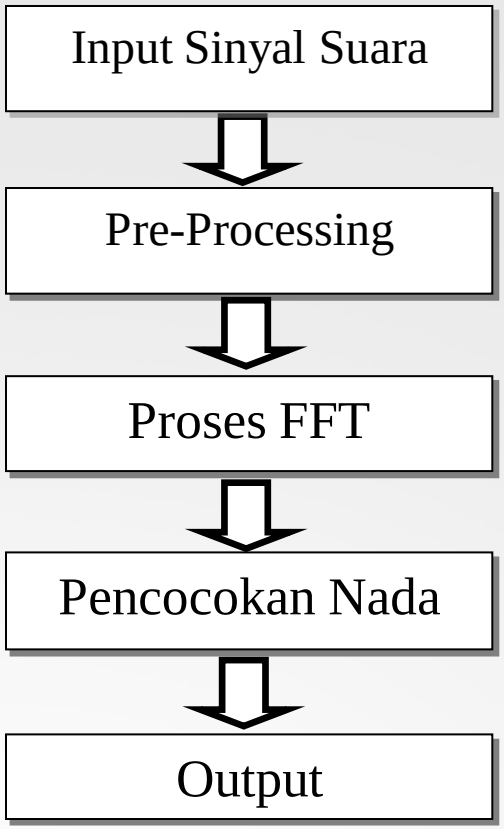
Tabel Frekuensi Nada kecapi oktaf pertama sampai oktaf ke-empat

| Nada | Pelog Degung (Hz) | | | | Madenda (Hz) | | | |
|---|---|---|---|---|---|---|---|---|
| | Oktaf | | | | Oktaf | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Da | 111 | 223 | 445 | 891 | 111 | 223 | 445 | 891 |
| Mi | 106 | 213 | 425 | 851 | 106 | 213 | 425 | 851 |
| Na | 84 | 169 | 338 | 675 | 95 | 189 | 379 | 758 |
| Ti | 73 | 147 | 294 | 588 | 75 | 150 | 301 | 601 |
| La | 70 | 140 | 281 | 561 | 70 | 140 | 281 | 561 |

# Analisis Proses Sound Recognition

- *Sound Recognition* merupakan sebuah teknologi terapan yang mengubah sinyal-sinyal digital yang berasal dari bunyi menjadi instruksi atau aksi tertentu.
- Di dalam istilah musik, *pitch*/nada berarti sama seperti frekuensi.
- Secara garis besar, cara kerja *sound recognition* ialah sinyal suara audio yang diterima melalui mikrofon diambil salah satu komponennya yaitu frekuensi. Frekuensi tersebut dapat direpresentasikan dalam bentuk spektrum. Dengan mentransformasikan sinyal dari domain waktu kedalam domain frekuensi, menggunakan rumus Fast Fourier Transform, akan diperoleh sekumpulan informasi tentang frekuensi yang terkandung dalam sinyal tersebut. Pemisahan ini dapat mencirikan suatu objek suara, sehingga suara dapat dikenali.

# Analisis Proses Sound Recognition (2)



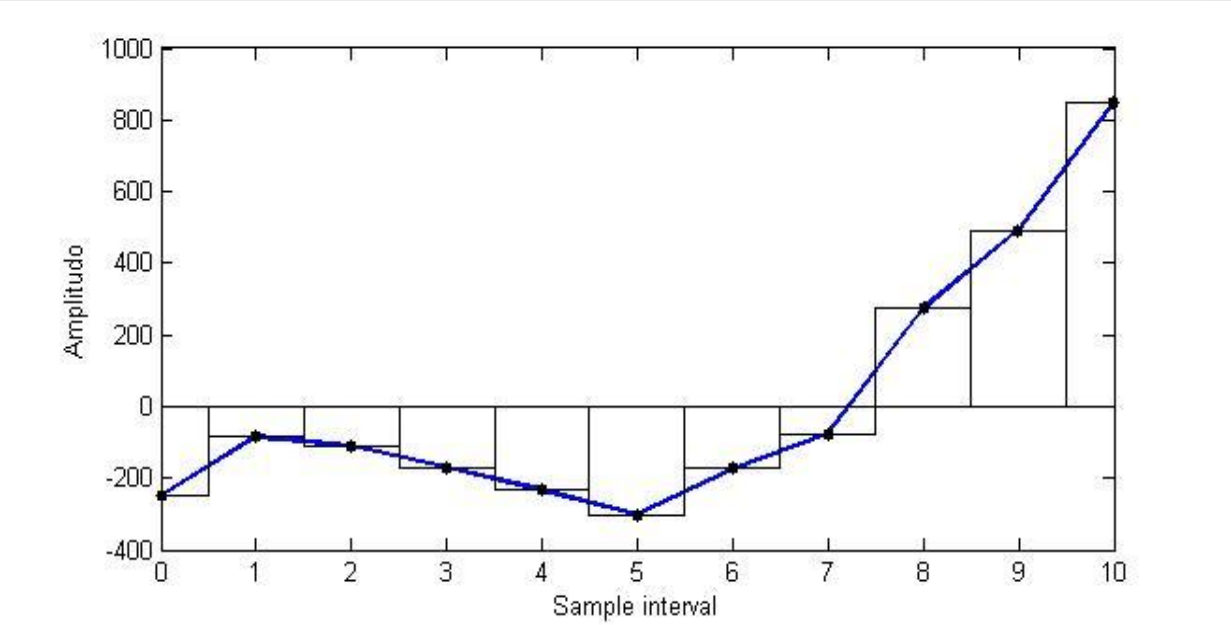

Gambar Diagram Blok Sistem Untuk Proses *Sound Recognition*

# Pre-prosessing

- **Sampling**
  - Data berupa sinyal suara diperoleh dengan cara merekam suara melalui mikrofon yang terdapat pada perangkat android.
  - Perekaman suara di dalam aplikasi menggunakan frekuensi sampling standar 8000Hz.
  - Suara yang direkam ini menggunakan 16 *bits/sample* dan 1 *channel* untuk *channel mono.*
  - Dalam analisis ini digunakan durasi rekaman yang diambil adalah selama 1 detik.

# Pre-prosessing (2)

- **Kuantisasi**
  - Proses ini adalah proses pengkonversian nilai analog ke dalam suatu nilai diskrit.




Gambar Input Sinyal Analog yang di Kuantisasi

| | |
|---|---|
| x(0) | -250 |
| x(1) | -85 |
| x(2) | -109 |
| x(3) | -170 |
| x(4) | -232 |
| x(5) | -302 |
| x(6) | -174 |
| x(7) | -77 |
| x(8) | 274 |
| x(9) | 492 |
| x(10) | 849 |

# Pre-prosessing (3)

- ***Coding***
  - *Coding*, pada proses ini, tiap nilai diskrit yang telah didapat, direpresentasikan dengan angka *binary* n-bit.
  - Cara konversinya adalah dengan membagi bilangan desimal dengan bilangan biner dengan memperhatikan hasil sisa pembagian.
  - Kode biner hasil *coding* tidak akan dimasukan dalam perhitungan proses selanjutnya, karena kode biner ini hanya dibaca oleh *processor* untuk pembacaan suatu sinyal digital.

10:43 PM

# Proses Fast Fourier Transform

- *Fast Fourier Transform* adalah suatu algoritma yang digunakan untuk merepresentasikan sinyal dalam domain waktu diskrit kedalam domain frekuensi.
- Dengan menggunakan algoritma Fast Fourier Transform (FFT) perhitungan DFT dapat dilakukan dengan cepat dan efisien.
- DFT dapat dituliskan sebagai berikut :

$$X[k] = \sum_{k=0}^{N-1} x[j] W_N^{nk}$$

Dimana $W = e^{-2\pi i / N}$

10:43 PM

# Proses Fast Fourier Transform

- Menghitung deret DFT secara langsung memerlukan operasi aritmetika sebanyak O($N^2$).
- Sebuah algoritma FFT hanya memerlukan operasi sebanyak O($N$ log $N$) untuk menghitung deret yang sama. Secara umum algoritma tersebut tergantung pada pemfaktoran $N$.

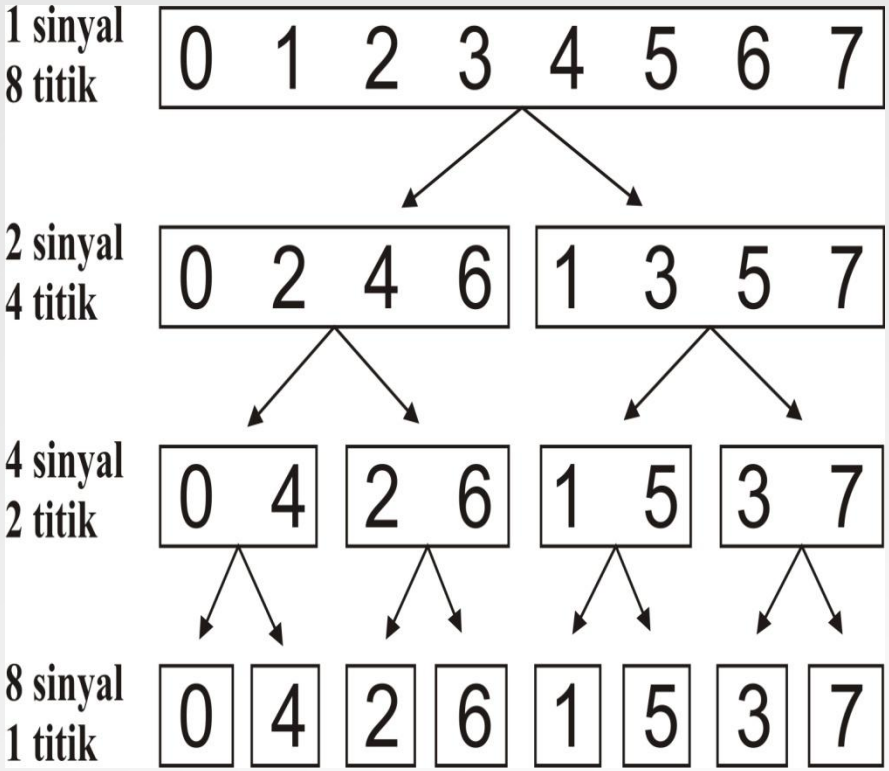
$$X[k] = \sum_{j=0}^{N-1} e^{\frac{2\pi i j k}{N}} x[j]$$

$$= \sum_{j=0}^{N/2-1} e^{\frac{2\pi i j k/N}{N}} x[2j] + \sum_{j=0}^{N/2-1} e^{\frac{2\pi i k(2j+1)/N}{N}} x[2+1]$$

$$= \sum_{j=0}^{N/2-1} e^{\frac{2\pi i k j}{N/2}} x[2j] + W_N^k \sum_{j=0}^{N/2-1} e^{\frac{2\pi i k j}{N/2}} x[2j+1]$$

$$= \sum_{j=0}^{N/2-1} x[2j]\, W_{N/2}^{kj} + W_N^k \sum_{j=0}^{N/2-1} x[2j+1]\, W_{N/2}^{kj}$$

# Proses Fast Fourier Transform (2)

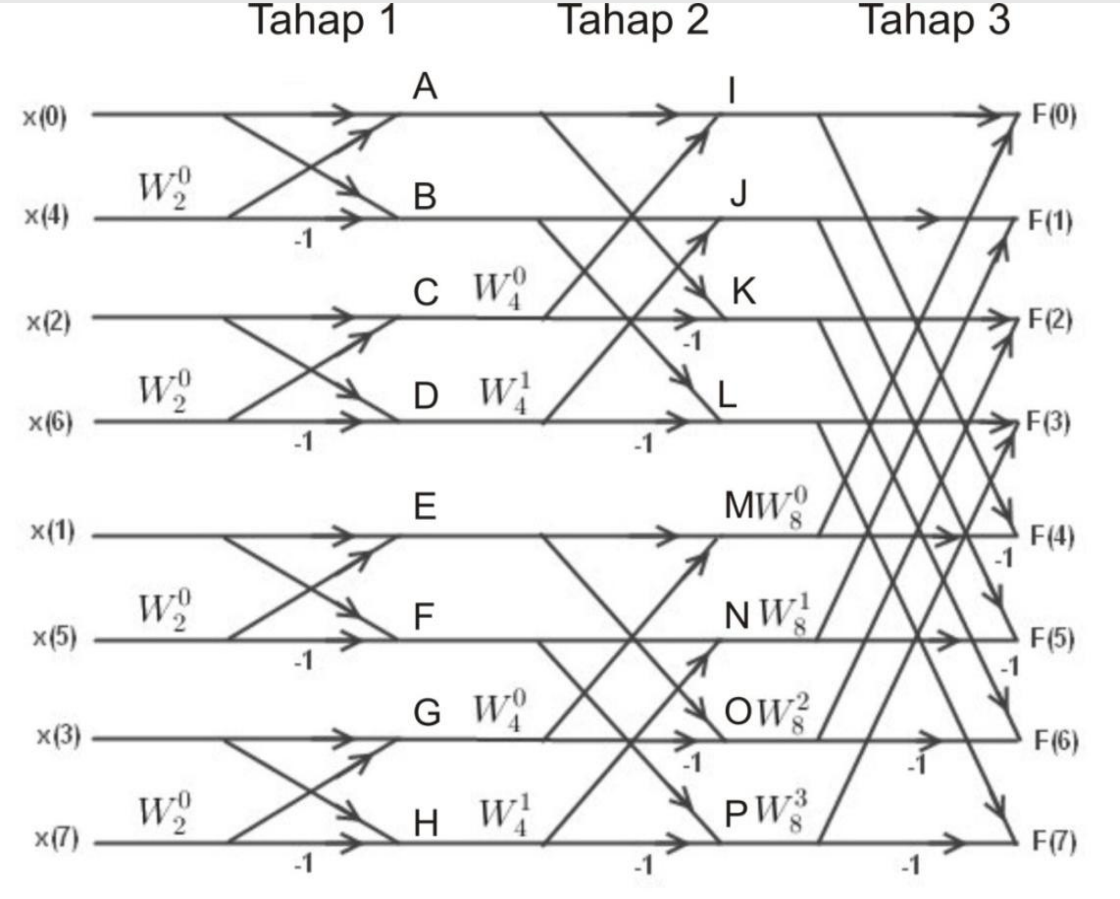



Gambar. Desimasi untuk 8 titik

| Urutan normal | | Urutan setelah pembalikan | |
|---|---|---|---|
| Desimal | Biner | Desimal | Biner |
| 0 | 000 | 0 | 000 |
| 1 | 001 | 4 | 100 |
| 2 | 010 | 2 | 010 |
| 3 | 011 | 6 | 110 |
| 4 | 100 | 1 | 001 |
| 5 | 101 | 5 | 101 |
| 6 | 110 | 3 | 011 |
| 7 | 111 | 7 | 111 |

Gambar. Pembalikan bit (Bit Reversal) untuk 8 titik

# Proses Fast Fourier Transform (3)




Gambar 3.8. Metode kupu-kupu 8-titik

# Proses Fast Fourier Transform (4)

$$F(0) = x(0) + W_2^0\,x(4) + W_4^0\,x(2) + W_4^0\,W_2^0\,x(6) + W_8^0\,x(1) + W_8^0 W_2^0\,x(5) + W_8^0\,W_4^0\,x(3) + W_8^0\,W_4^0\,W_2^0\,x(7)$$

$$F(1) = x(0) - W_2^0\,x(4) + W_4^1\,x(2) - W_4^0\,W_2^0\,x(6) + W_8^1\,x(1) - W_8^1 W_2^0\,x(5) + W_8^1\,W_4^0\,x(3) + W_8^1\,W_4^0\,W_2^0\,x(7)$$

$$F(2) = x(0) + W_2^0\,x(4) - W_4^0\,x(2) + W_4^0\,W_2^0\,x(6) + W_8^2\,x(1) + W_8^2 W_2^0\,x(5) - W_8^2\,W_4^0\,x(3) - W_8^2\,W_4^0\,W_2^0\,x(7)$$

$$F(3) = x(0) - W_2^0\,x(4) - W_4^1\,x(2) - W_4^1\,W_2^0\,x(6) + W_8^3\,x(1) - W_8^3 W_2^0\,x(5) - W_8^3\,W_4^1\,x(3) - W_8^3\,W_4^1\,W_2^0\,x(7)$$

$$F(4) = x(0) + W_2^0\,x(4) + W_4^0\,x(2) + W_4^0\,W_2^0\,x(6) - W_8^0\,x(1) - W_8^0 W_2^0\,x(5) - W_8^0\,W_4^0\,x(3) - W_8^0\,W_4^0\,W_2^0\,x(7)$$

$$F(5) = x(0) - W_2^0\,x(4) + W_4^1\,x(2) - W_4^1\,W_2^0\,x(6) - W_8^1\,x(1) + W_8^1 W_2^0\,x(5) - W_8^1\,W_4^1\,x(3) + W_8^1\,W_4^1\,W_2^0\,x(7)$$

$$F(6) = x(0) + W_2^0\,x(4) - W_4^0\,x(2) + W_4^0\,W_2^0\,x(6) - W_8^2\,x(1) - W_8^2 W_2^0\,x(5) + W_8^2\,W_4^0\,x(3) + W_8^2\,W_4^0\,W_2^0\,x(7)$$

$$F(7) = x(0) - W_2^0\,x(4) - W_4^1\,x(2) - W_4^1\,W_2^0\,x(6) - W_8^3\,x(1) + W_8^3 W_2^0\,x(5) + W_8^3\,W_4^1\,x(3) + W_8^3\,W_4^1\,W_2^0\,x(7)$$

Dimana $W_N^k = e^{\frac{-j2\pi k}{N}} = \cos\left(\frac{-2\pi k}{N}\right) + \mathrm{j}\,\sin\left(\frac{-2\pi k}{N}\right)$

Misal $W_2^0 = \cos\left(\frac{-2\pi(0)}{2}\right) + j\sin\left(\frac{-2\pi(0)}{2}\right) = 1 + 0j$

10:43 PM

# Proses Fast Fourier Transform (5)

Dari hasil kuantisasi sebelumnya maka dapat diambil data suara hasil sinyal diskrit dengan diwakili nilai x(k) sebagai contoh kita gunakan 8 sample data :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -250 | -85 | -109 | -170 | -232 | -302 | -174 | -77 |
| x(0) | x(1) | x(2) | x(3) | x(4) | x(5) | x(6) | x(7) |

Subtitusikan sample data ini kedalam persamaan sebelumnya:

$$F(0) = x(0) + W_2^0\,x(4) + W_4^0\,x(2) + W_4^0\,W_2^0\,x(6) + W_8^0\,x(1) + W_8^0 W_2^0\,x(5) + W_8^0\,W_4^0\,x(3) + W_8^0\,W_4^0\,W_2^0\,x(7)$$

$$= -250 + (1 * (-232)) + (1 * (-109)) + (1 * 1 * (-174)) + (1 * (-85)) + (1 * 1 * (-302)) + (1 * 1 * (-170)) + (1 * 1 * 1 * (-77))$$

$$= -1399 + 0j$$

Lakukan hal yang sama kepada ke-tujuh data sinyal lainnya. Sehingga menghasilkan nilai sebagai berikut:

$F(1) = 134{,}8 + 130{,}15j$ $F(2) = -524 + 294j$ $F(3) = 3{,}2 - 261{,}8j$

$F(4) = -131 + 0j$ $F(5) = -105{,}68 + 22{,}68j$ $F(6) = -894 + 162j$

$F(7) = 39{,}2 - 63{,}8j$

# Proses Fast Fourier Transform (6)

- Magnitude pada frekuensi dapat dihitung dengan rumus :

  Magnitude $F(0) = |[-1399^2 + 0^2]| = 1399$

- Maka magnitude dari $F(0) = 1399$, dengan cara yang sama lakukan kepada ke-tujuh data sinyal lainnya. Sehingga menghasilkan nilai magnitude:

  $F(0) = 1399$ $F(1) = 187{,}38$ $F(2) = 621$ $F(3) = 261{,}82$

  $F(4) = 131$ $F(5) = 108{,}09$ $F(6) = 864{,}32$ $F(7) = 74{,}88$

- Bagian frekuensi positif terkecil dimulai dari *F*(1) sampai *F*(N), maka data sinyal yang memiliki magnitude terbesar terdapat pada F(6), maka diperoleh :
  - Frekuensi = 6 * Fs / N
    = 6 * 8 / 8
    = 6 Hz

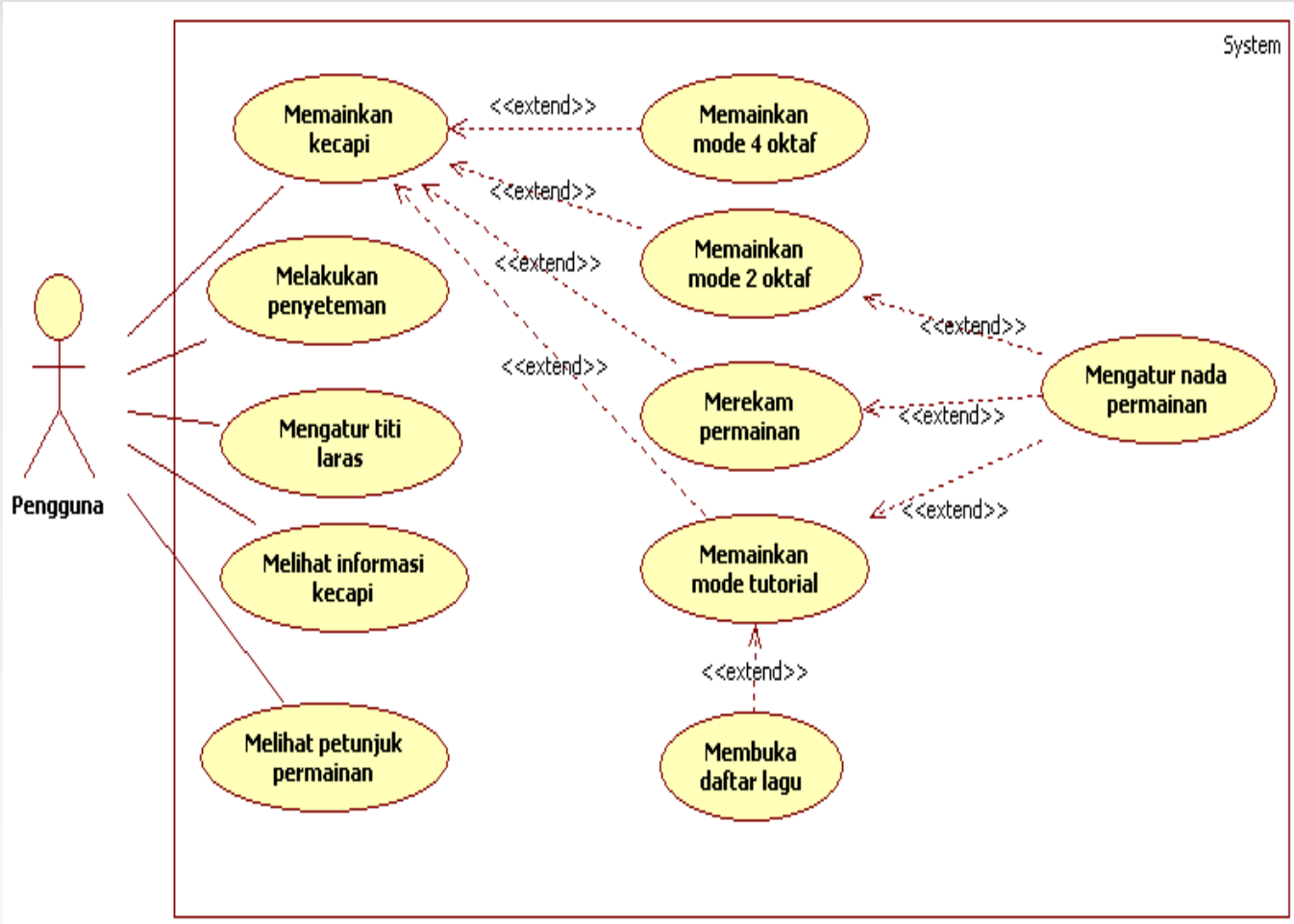
  Fs = Sample rate

  N = Jumlah sample sinyal

# Pencocokan Frekuensi

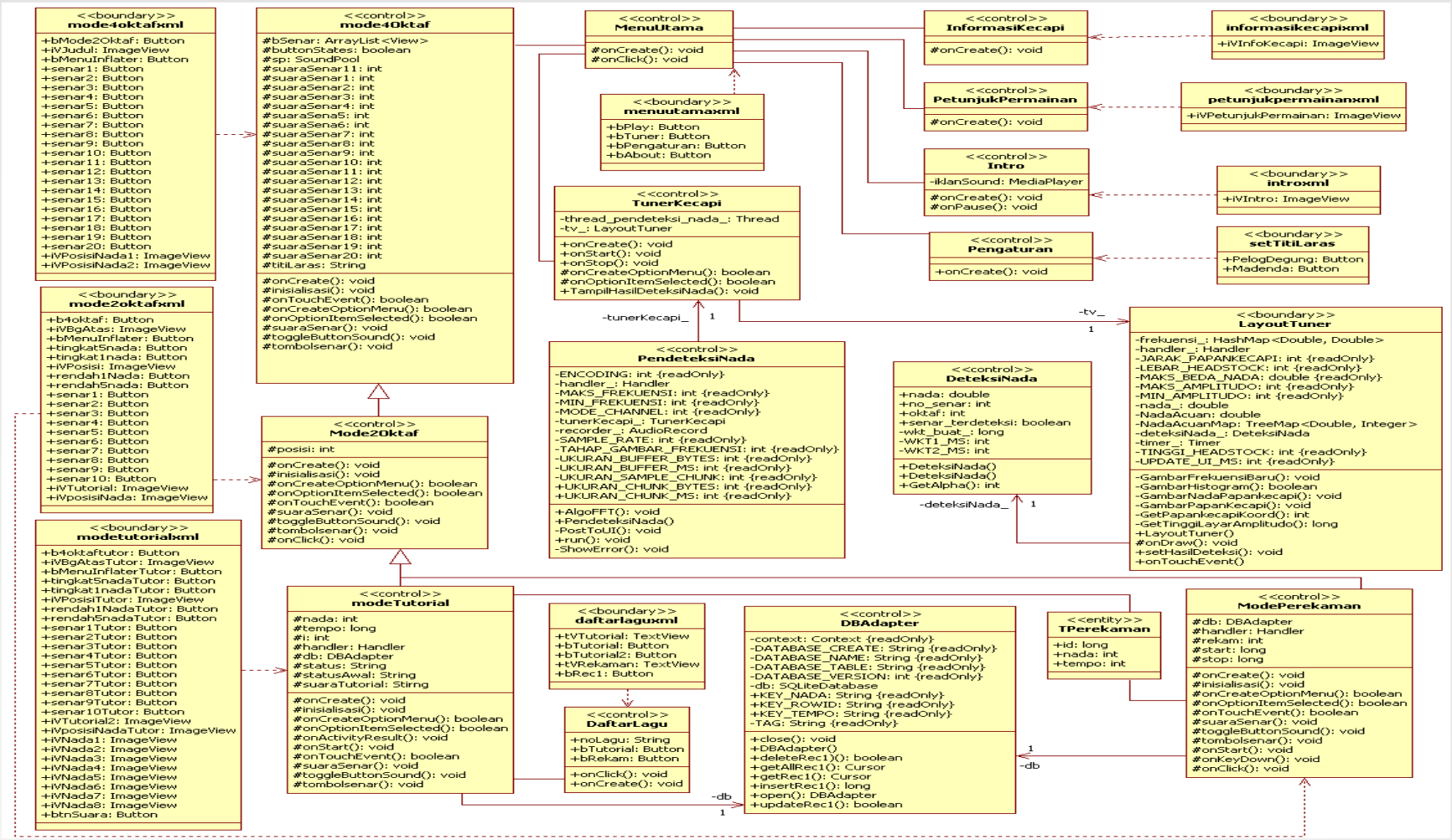
- Untuk memudahkan pencocokan, nilai frekuensi nada acuan digunakan sebagai pembanding terhadap frekuensi yang dihasilkan.
- Untuk memudahkan pencocokan frekuensi, diberikan nilai maksimal beda nada sebesar 10 Hz dari nada acuan.
- Jika nada yang dihasilkan memiliki selisih kurang dari 10 Hz dari nada acuan maka akan ditampilkan posisi nada tersebut berserta jumlah selisih frekuensi masukan dengan nada acuan.

# Use Case Diagram



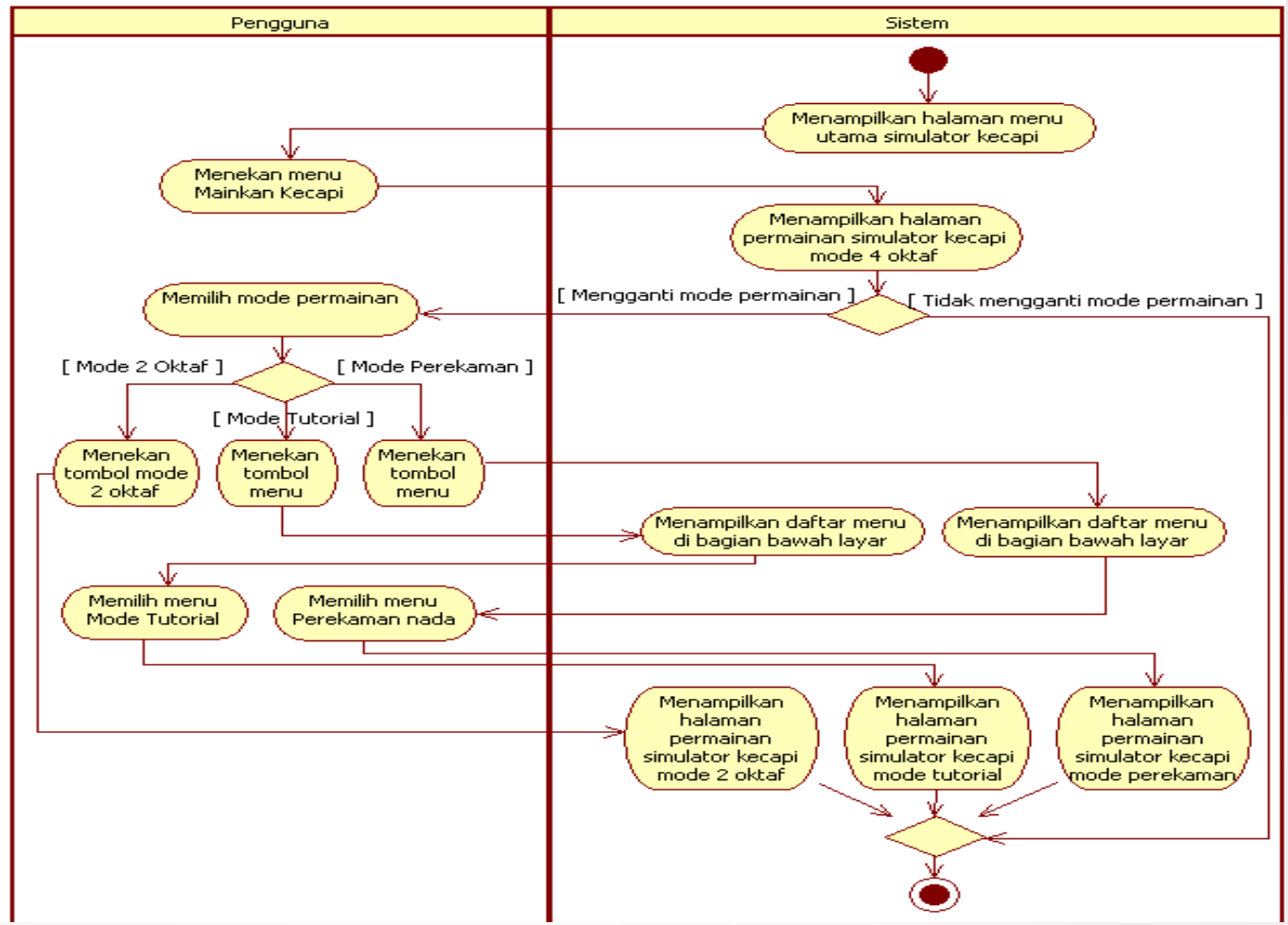

Gambar. Use Case Simulator Kecapi

# Class Diagram



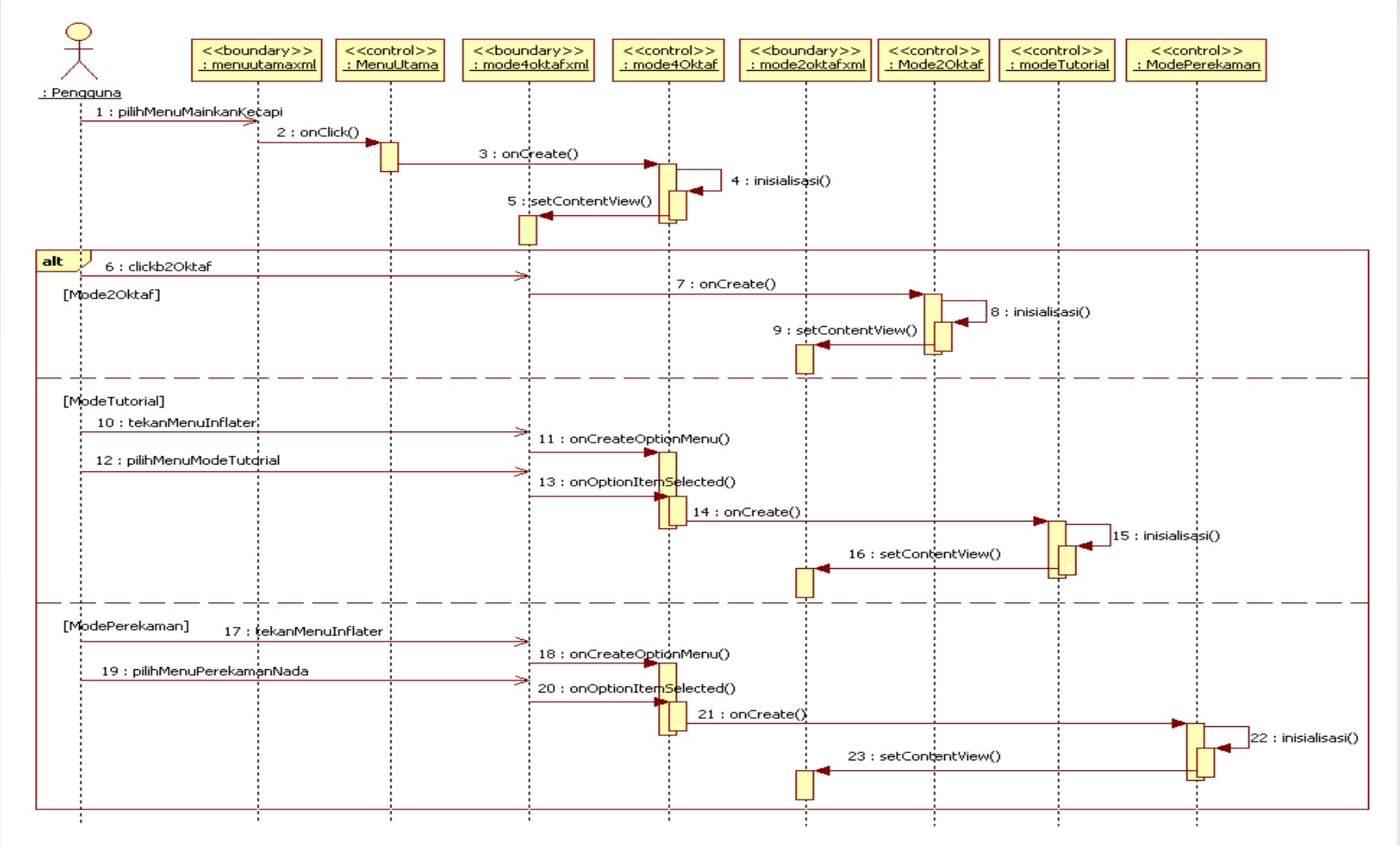

Gambar 3.10. Class Diagram Aplikasi Simuator Kecapi

# Activity Diagram Memainkan Kecapi




Gambar . *Activity Diagram* Memainkan Kecapi

# Sequence Diagram Memainkan Kecapi




Gambar *Sequence Diagram* Memainkan kecapi

# Implementasi dan Pengujian



# Perangkat Keras Pembangun

1. Spesifikasi perangkat keras komputer
   - Intel Pentium Core 2 Duo 2.00 GHz
   - Memori 2 GB
   - VGA 128 MB
   - *Harddisk* 160 GB
   - Monitor
   - *Mouse*
   - *Keyboard*
   - *Microphone*

# Perangkat Keras Pembangun (2)

2. Spesifikasi perangkat keras *Smartphone*

   Adapun *smartphone* yang digunakan adalah Huawei Ideos S7 Slim dengan spesifikasi sebagai berikut:

   - Sistem Operasi : Android 2.2 (*Froyo*)
   - CPU : 1 GHz *Scorpion*
   - Memory : 512MB Ram, 512Mb ROM
   - Dimensi layar : 480x800
   - Fitur *Audio Record*



# Perangkat Lunak Pembangun

- Sistem Operasi *Windows XP Profesional SP 2*
- *Eclipse Helios* 3.6.2
- JDK (*Java Development Kit*) dan JRE (*Java Runtime Environment*)
- Android SDK (*Software Development Kit*)
- ADT (*Android Development Tools*)
- DBMS SQLite

# Kesimpulan Pengujian *Alpha*

- Secara fungsional semua proses pada aplikasi simulator kecapi berbasis Android yang dibangun sudah berjalan sesuai dengan yang diharapkan baik itu dari output berdasarkan input tertentu yang diberikan maupun dari segi validasi dan proses penanganan kesalahan.

# Kesimpulan Pengujian *Beta*

- *user friendly*
- mudah dioperasikan
- sangat membantu dalam memperluas informasi mengenai alat musik kecapi sebagai alat musik kebudayaan Jawa Barat
- sangat membantu dalam mempopulerkan alat musik kecapi
- memberikan sarana menuangkan ide dalam permainan kecapi
- sangat membantu dalam proses penyeteman nada alat musik kecapi.

# Kesimpulan

- Aplikasi simulator kecapi ini membantu dalam memperluas informasi mengenai alat musik kecapi sebagai alat musik kebudayaan Jawa Barat.
- Aplikasi simulator kecapi ini membantu dalam mempopulerkan alat musik kecapi
- Aplikasi simulator kecapi ini memberikan sarana menuangkan ide dalam permainan kecapi
- Aplikasi simulator ini membantu dalam proses penyeteman nada pada alat musik kecapi

# Saran

- Jumlah lagu pada daftar lagu tutorial perlu diperbanyak sehingga menambah cara memainkan lagu pada aplikasi simulator kecapi
- Media penyimpanan lagu hasil perekaman perlu dibuat lebih dinamis sehingga pengguna dapat merekam lebih banyak lagu
- Untuk selanjutnya disarankan aplikasi ini dapat berjalan pada platform selain android misalnya blackberry atau iPhone

# Demo Program

## Kecapi Simulator




4 Oktave Mode

2 Oktave Mode

Tuning

Recording

Tutorial