

### **III. PRAKTIKUM 2 – Proses Akuisisi Data Audio Dengan Memakai USB-Soundcard**

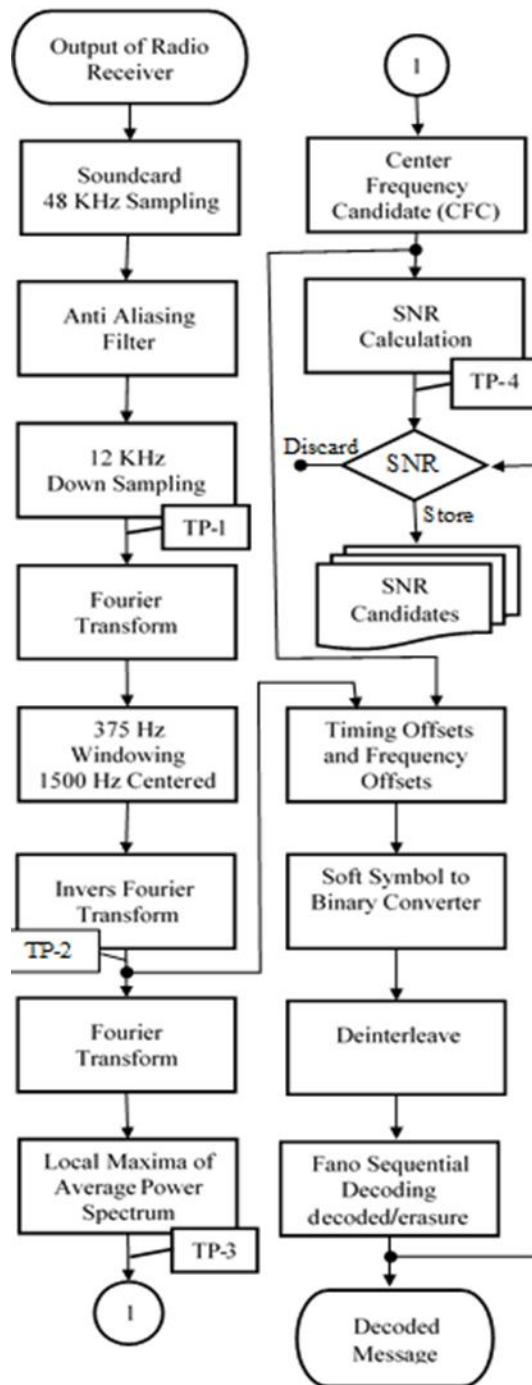
#### **III.1 PENDAHULUAN**

Proses panjang untuk mendekode data WSPR yang dikirim dari tempat yang sangat jauh melalui kanal HF dapat dilihat pada blok diagram Gambar III-1. Praktikum 2 ini dimulai dengan mempelajari cara memanfaatkan Raspberry Pi dan periferalnya. WSPR adalah sistem komunikasi digital yang sensitif terhadap pewaktu/clock. Oleh karena itu diperlukan sinkronisasi waktu antara pemancar dan penerima yang dapat didasarkan pada stempel waktu Internet dengan hanya memanggil library standar C yang berisi deklarasi fungsi waktu dan tanggal untuk memberikan akses standar ke penanda waktu/tanggal. Hal kedua yang akan diamati adalah konsep dasar akuisisi sinyal dengan menggunakan USB-Soundcard di lingkungan library ALSA. Proses ini akan diperkenalkan kepada mahasiswa melalui pengamatan program C yang diberikan dalam modul praktikum. Pertama-tama siswa diperkenalkan dengan inisialisasi perangkat audio untuk mengambil data, dengan memanggil Application Program Interface (API) ALSA. Para mahasiswa harus mengenali tujuan setiap baris sintaks C yang diberikan oleh modul praktik. Jumlah bit per sampel, format data, ukuran buffer, struktur sampel, frekuensi sampling, struktur frame sampel, aktivasi/handling perangkat I/O, adalah parameter yang akan diinisialisasi dengan melewati parameter-parameter tersebut melalui sintaksis C program.

#### **III.2 KISI-KISI PENTING**

Kisi-kisi penting sebelum masuk ke pemrograman dalam sistem penerima WSPR adalah sbb:

- WSPR memancar secara sinkron setiap masuk ke hitungan menit genab.
- Data WSPR selesai memancar pada detik ke 110,6.
- Sehingga RASPI punya waktu untuk dekode hanya  $(120-110,6=9,4)$  detik agar siap menerima paket data berikutnya.
- Data baseband masuk ke RASPI via USB soundcard dengan frekuensi sampling 12KHz.
- Buffer 114 detik (180 blok-simbol) untuk mendeteksi sinyal WSPR 162 blok-simbol atau 110,6 detik.
- Terdapat  $(12000*114) = 1.368.000$  sampel.
- Format PCM 2 byte short integer per sampel.



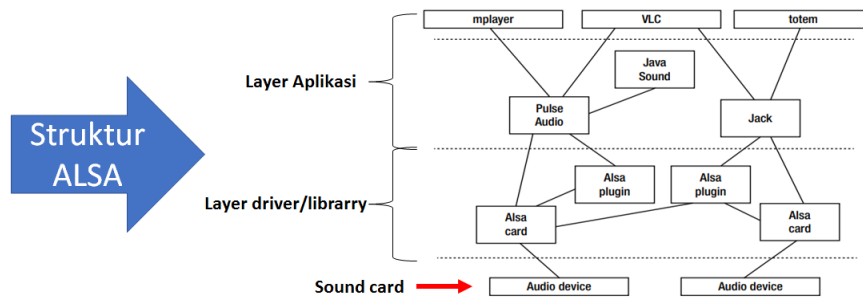
Gambar III-1: Blok diagram proses panjang untuk mendekode sinyal power lemah WSPR

Dewasa ini dapat dikatakan bahwa library ALSA adalah satu satunya library yang ada pada sistem operasi Linux untuk keperluan API (Application Program Interface) dalam hal pengembangan pengaturan sistem Soundcard. Semua aplikasi audio seperti JACK, Pulse, JavaSound, Totem dan Mplayer adalah aplikasi sistem pengolahan/perekaman audio profesional yang mengandalkan library ALSA untuk berhubungan dengan perangkat keras Soundcard yang ditempelkan pada

komputer prosesor tersebut. Kaidah hirarkhi interkoneksinya dapat dilihat pada Gambar III-2 dibawah ini.

## Mengontrol Sound Card dengan ALSA-Lib.

- ALSA = Advanced Linux Sound Architecture
- URL= <https://www.alsa-project.org/alsa-doc/alsa-lib/pcm.html>



Gambar III-2: Kaidah interkoneksi antara library ALSA dengan aplikasi dan soundcard pendukungnya dalam membangun sistem audio digital yang canggih.

### III.3 LANGKAH PERCOBAAN

1. Siapkan modul Raspberry Pi untuk siap dioperasikan.
2. Pastikan bahwa Raspberry Pi sudah mempunyai sistem perangkat lunak sinkronisasi waktu, coba dengan command "timedatectl" dan yakinkan bahwa respon Raspi sudah benar (waktu yang ditunjukkan sudah menunjuk angka yang benar).  
%> timedatectl \r
3. Masuk ke direktori "PRAKTIKUM-REC"  
%>cd PRAKTIKUM-REC \r
4. Jalankan program untuk memeriksa kesiapan USB-Soundcard.  
%> ./hw-record \r  
Amatilah respon dari Raspi, catat kemampuan format, frekuensi sampling dan jumlah jumlah kanal yang dapat ditangani oleh USB-Soundcard untuk operasi merekam suara.  
%> ./hw-play \r  
Amatilah respon dari Raspi, catat kemampuan format, frekuensi sampling dan jumlah jumlah kanal yang mungkin untuk dikerjakan oleh USB-Soundcard untuk operasi playback suara.
5. Jika Raspi sudah terhubung ke Radio (radio belum dinyalakan), konsultasikan ke dosen pengampu praktikum bahwa sudah tersedia sinyal WSPR di udara (bisa secara lokal dibangkitkan oleh laboratorium sendiri). Tanyakan frekuensi kerja sinyal sumber WSPR yang mungkin/sedang memancar.

6. Nyalakan radio (dengan konsultasi terlebih dahulu kepada dosen pengampu praktikum), atur frekuensi sesuai arahan dosen pengampu.
7. Kembali ke modul Raspberry-Pi, jalankan program “rekam-time”.  
`%> ./rekam-time \r`  
 Selanjutnya program akan menunggu sampai menit genap untuk memulai proses perekaman sinyal WSPR sepanjang 114 detik kedepan.
8. Perhatikan file “raw” yang dihasilkan oleh proses perekaman ini. Dengan melihat besarnya byte yang ditampilkan pada file tersebut, menurut Anda apakah sudah benar.
9. Sebelum mengerjakan tugas dan menulis laporan praktikum, silahkan perhatikan 3 gambar highlight sintak program seperti Gambar III-3 dibawah ini.

## Konfigurasi ALSA

Fungsi-fungsi dalam Library ALSA ini Akan mengembalikan nilai “0” jika Sukses dan mengembalikan nilai lain Jika gagal.

- Mengenali soundcard dalam moda merekam suara →  
`snd_pcm_open(&capture_handle, devcID, SND_PCM_STREAM_CAPTURE, 0)`
- Inisiasi parameter →  
`snd_pcm_hw_params_any(capture_handle, hw_params)`
- Tipe akses →  
`snd_pcm_hw_params_set_access(capture_handle, hw_params, SND_PCM_ACCESS_RW_INTERLEAVED)`
- Format data →  
`snd_pcm_hw_params_set_format(capture_handle, hw_params, format)`

## Konfigurasi ALSA... lanjutan

- Frekuensi sampling →  
`snd_pcm_hw_params_set_rate_near(capture_handle, hw_params, &rate, 0)`
- Jumlah saluran minimum →  
`snd_pcm_hw_params_get_channels_min(hw_params, &min);`
- Jumlah saluran maksimum →  
`snd_pcm_hw_params_get_channels_max(hw_params, &max);`
- Setting jumlah kanal →  
`snd_pcm_hw_params_set_channels(capture_handle, hw_params, min)`

## Konfigurasi ALSA... lanjutan

- Persiapan soundcard pengambilan data →  
`snd_pcm_prepare(capture_handle)`
- Persiapan buffer memori untuk simpan suara 114 detik →  
`buffer = malloc(buffer_frames_48 *  
snd_pcm_format_width(format)/8*2);`
- Buffering untuk simpan suara 114 detik →  
`if((err=snd_pcm_readi(capture_handle, buffer,  
buffer_frames_48)) != buffer_frames_48){  
Exit(-1)  
}`

Gambar III-3: Highlight sintak pemrograman untuk merekam suara sepanjang 114 detik

### III.4 TUGAS DAN LAPORAN PRAKTIKUM

Tunjukkan sintak atau sekumpulan sintak yang :

1. mengatur pewaktu untuk start perekaman.
  2. menentukan besarnya bufer memori sementara untuk data 114 detik suara.
  3. menentukan moda capture/perekaman atau moda playback.
  4. menentukan lebar frame. Sebutan lain dari lebar frame adalah  
"mono/stereo/quadro" atau jumlah kanal input, mono=1, stereo=2, quadro=4, dst.
  5. menentukan besar frekuensi sampling.
  6. menentukan format sampel PCM 16 bit.
  7. melaksanakan proses penyimpanan permanen data 114 detik.
- 
-