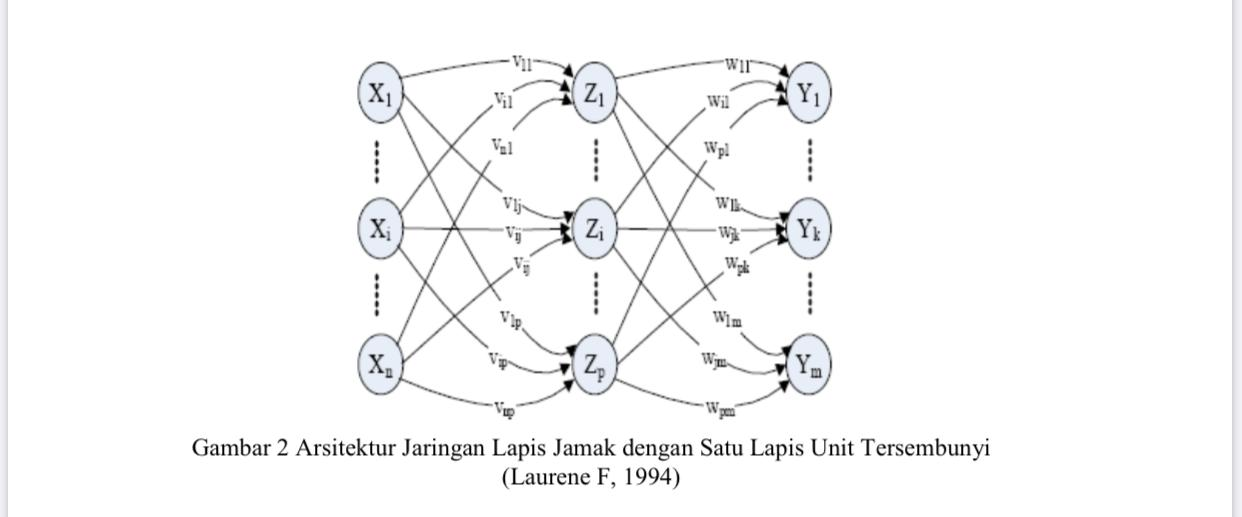
1. **Metode yang Digunakan**

Pada laporan ini kami akan menggunakan metode Jaringan Syaraf Tiruan (JST) Backpropagation. JST Backpropagation biasanya digunakan untuk melakukan pengenalan pola, klasifikasi, pengolahan citra dan pengambilan keputusan. Metode ini merupakan JST pembelajaran yang supervised, yang artinya data pembelajaran terdiri dari vektor pasangan input dan target (output yang diharapkan).

JST Backpropagation memiliki dua tahap yaitu *forward pass* (perambatan maju) dan *backward pass* (perambatan mundur). *Forward pass* adalah mencari nilai hidden layer dan  output sedangkan *backward pass* adalah membandingkan output dengan target. perambatan mundur. Selama *forward pass*, tiap unit masukan (Xi) menerima sebuah masukan sinyal ini ke tiap-tiap tersembunyi ini kemudian menghitung aktivasinya dan mengirimkan sinyalnya (Zj) ke tiap unit keluaran.



Arsitektur Jaringan Lapis Jamak dengan Satu Lapis Unit Tersembunyi

Proses identifikasi suara dengan JST Backpropagation melalui 3 tahap yaitu *forward pass, backward pass,* dan perubahan bobot. Output dari sistem adalah suara yang terdeteksi baik dikenali sebagai target maupun tidak dikenali. Proses pelatihan dilakukan sampai jaringan memperoleh nilai *error*  target sekecil mungkin, Setelah syarat tersebut tercapai, maka dilakukan penyimpanan bobot dari setiap ruas pada jaringan. Bobot - bobot tersebut nantinya akan digunakan pada proses pengujian.

**Cara menghitung forward pass**

Dengan langkah - langkah sebagai berikut :

1. Menentukan nilai input, bias , learning rate, weight dan target
2. Menghitung hidden layer hidden 1 dan hidden 2 (h1 dan h2) dengan rumus :

h1 = (i1 \* w1) + (i2 \* w2) + b

kemudian ketika sudah menemukan hasilnya maka selanjutnya menghitung fungsi sigmoid dari hasil h1 dan h2 dengan rumus :

Out h1 = 1/(1 + e ^ -h1) dengan hasil e = 2.71828183

1. Menghitung  output layer 1 dan 2 (O1 dan O2) dengan rumus :

O1 = (Out H1 \* w5) + (Out H2 \* w6) + b2

kemudian dihitung menggunakan fungsi sigmoid juga seperti menghitung hidden layer dengan rumus :

Out O1 = 1/(1 + e ^ -O1) -> e = 2.71828183

1. Menghitung nilai error total (E1 dan E2) dengan rumus seperti menghitung E1 kemudian jumlahkan hasil E1 dan E2. Untuk mencari E1 dengan rumus berikut :

E1 = 1/2 (T1 - Out O1)^2

**Cara menghitung backward pass**

Dengan langkah - langkah sebagai berikut :

1. Pada bagian ini kita update weight pada output layer  dengan rumus berikut :

    Etotal    Etotal OutO1   O1

rumus =  ------ = --------\* --------- \* ----

    Ow5   Out O1   O1 w5

untuk mencari hasilnya maka kita harus menghitung satu persatu - satu rumusnya dengan cara sebagai berikut :

  Etotal          1

                          --------- = 2 \* - (T1 - OutO1)^2-1 \* -1 + 0

  OutO1          2

          OutO1

      = ---------- = OutO1 (1 - OutO1)

  O1

O1

      = ------ = 1 \* OutH1 \* w5^(1-1) + 0 + 0

w5

Kemudian setelah menemukan hasilnya maka dimasukkan kerumus awal dan dikalikan.

langkah terakhir yaitu melakukan update pada weight :

        Etotal

w5^+ = w5 - learning rate \* --------

          w5

Dan melakukan hal yang sama terhadap w6^+, w7^+ dan w8^+

1. Mengupdate weight pada hidden layer

    Etotal   Etotal   OutH1 H1

rumus = -------- = -------- \* -------- \* ----

    w1   Out H1     H1 w1

dan melakukan hal yang sama dengan menghitung rumusnya satu per satu

      Etotal

mencari  = --------

        w1

dengan 2 langkah yaitu sebagai berikut :

 langkah 1 : E1      E1 OutH1

      ---- = ------ \* -------

      O1 OutO1  O1

    O1

catatan :--------- = w5

                OutH1

langkah 2 :  E1 E1   O1

      --------- = --- \*   ------ dan mencari E2 nya

      OutH1 O1   OutH1

Etotal       E1       E2

-------- = --------- \* ---------

OutH1    OutH1 OutH2

OutH1

--------- = OutH1(1-OutH1)

    H1

Setelah mendapatkan hasil semuanya maka dimasukkan rumus diatas dengan mengkalikan semua hasilnya.

Kemudian mengupdate weight seperti sebelumnya menggunakan rumus dibawah ini

      Etotal

w1^+ = w1 - learning rate \* ------  → terhadap w2^+, w3^+ dan w4^+

          w1